MEMORANDUM
RM-5762-PR
DECEMBER 1968

CURVES: A FIVE-FUNCTION CURVE-FITTING COMPUTER PROGRAM

H. E. Boren, Jr.

PREPARED FOR:

UNITED STATES AIR FORCE PROJECT RAND

The RAND Corporation
SANTA MONICA • CALIFORNIA

90

MEMORANDUM
RM-5762-PR
DECEMBER 1968

CURVES: A FIVE-FUNCTION CURVE-FITTING COMPUTER PROGRAM

H. E. Boren, Jr.

This research is supported by the United States Air Force under Project RAND—Contract No. F44620-67-C-0045—monitored by the Directorate of Operational Requirements and Development Plans, Deputy Chief of Staff, Research and Development, Hq USAF. Views or conclusions contained in this study should not be interpreted as representing the official opinion or policy of the United States Air Force.

The RAND Corporation

This study is presented as a competent treatment of the subject, worthy of publication. The Rand Corporation vouches for the quality of the research, without necessarily endorsing the opinions and conclusions of the author.

Published by The RAND Corporation

PREFACE

The computer program (CURVES) described in this Memorandum was developed in support of estimating-relationship research efforts being conducted in the RAND Cost Analysis Department. This program represents a compilation of various parts of existing programs written by the author, with modifications being included where necessary. The author makes no claim to originality or to efficiency of operation with regard to the program. The main purpose of writing such a program was to have available for cost analysts an easily workable, user-oriented, curve-fitting computer program--especially adapted to handle the mathematical functions most commonly used in the development of estimating relationships.

SUMMARY

This Memorandum describes a FORTRAN-IV curve-fitting comput program that has been developed within the RAND Cost Analysis Department. The program makes least-squares determinations of the parameters of any of five mathematical functions selected by the user, given a set of observations on the dependent and independent variables of interest. The functions available in the program are the line, parabola, power, asymptotic-power, and exponential. Up to three independent variables may be used for the line and power functions. Also, the Y-intercept may be specified for the line, parabola, or asymptotic-power function.

A discussion of the characteristics of the functions is presented in Section I, including an examination of those nonlinear functions that require special methods for solution. Also included is a brief discussion of the statistics used in the program. Specific details on the operation of the program are presented in Section II. This section also treats the options available to the user. Program outputs are discussed in Section III. For the benefit of the reader, sample outputs from two runs are shown.

Mathematical considerations relating to nonlinear-least-squares solutions are treated in Appendices A and B. A listing of the FORTRAN-IV computer program is presented in Appendix C.

CONTENTS

FIGURES

TABLES

I. INTRODUCTION

PROGRAM DESCRIPTION

A FORTRAN-IV curve-fitting computer program (CURVES) has been written by the author that makes least-squares determinations of the parameters of any of five types of functions, given a set of observations on the dependent and independent variables of interest. These functions are commonly used in the derivation of cost analysis estimating relationships, and are: (a) line, (b) parabola, (c) power, (d) asymptotic-power, and (e) exponential. They are described in detail in subsequent parts of this section. Standard statistics relating to "goodness-of-fit" measures are also calculated in the program. No predictive statistics are included, however, because of the difficulty of obtaining such statistics for the nonlinear functions--the latter three above.* Consequently, the program is intended essentially for curve-fitting.

The CURVES program can handle up to 200 data points for each regression and is so structured that if a set of data cards contains data for several separate regressions, that set needs to be entered only once. This obviates the need for duplicating such input data decks for each regression run. A variable-format procedure is provided the user so that data may be entered in any order on the input cards. Also, an option is provided to allow the user to specify the Y-intercept value (regression constant) for the line, parabola, and asymptotic-power functions.

The program is written completely in FORTRAN-IV, using A4 formats for all alphanumeric information. No matrix-inversion or other subprograms are used. All solutions are made through either standard, algebraic methods for the linear and parabolic cases or through iterative methods for the other cases. Consequently, the program should be readily adaptable to other computer systems.

---

*In this Memorandum, a linear function is defined as one which is linear with respect to all of its parameters. Under this definition, the parabola is considered to be a linear function.

FUNCTION TYPES

The functions available in this program were chosen principally on the basis of their application to the derivation of cost analysis estimating relationships. They are:

1. Line (containing up to three independent variables)*

$$Y = A + B \cdot (X1),$$

$$Y = A + B \cdot (X1) + C \cdot (X2),$$

$$Y = A + B \cdot (X1) + C \cdot (X2) + D \cdot (X3)$$

Y-intercept (A) may be specified.

2. Parabola

$$Y = A + B \cdot (X1) + C \cdot (X1)^2.$$

3. Power (containing up to three independent variables)

$$Y = A \cdot (X1)^B,$$

$$Y = A \cdot (X1)^B \cdot (X2)^C,$$

$$Y = A \cdot (X1)^B \cdot (X2)^C \cdot (X3)^D.$$

4. Asymptotic-Power

$$Y = [A \cdot (X1)^B] + C$$

Y-intercept (C) may be specified.

5. Exponential

$$Y = e^{[A + B \cdot (X1)]},$$

*As in FORTRAN notation in the program, the independent variables are represented by X1, X2, and X3, respectively. When only one independent variable is considered, X1 is used. Also, as explained later, only positive values are considered for all variables.

where

Y = dependent variable,

X1, X2, X3 = independent variables,

A, B, C, D = parameters to be determined by least-squares methods,

e = constant ≈ 2.71828.

FUNCTION CHARACTERISTICS

Examples of some of the types of curves that can be obtained from the five functions are shown in Fig. 1 for a one-independent-variable case.

Line

The linear form is the most simple of the forms treated here. Its characteristics are well known and, in the opinion of the author, need no further elaboration. The user has the option of using up to three independent variables and also the option of specifying the Y-intercept (A).

Parabola

Sometimes the parabolic function is used to represent points that lie along a curve having a Y-intercept (including zero). However, one must be aware that since this function is actually a polynomial of degree 2, its curve always has a maximum or minimum point (vertex). This means that the effect of the independent variable (X1) on the dependent variable (Y) is reversed once this point is traversed. Again, the user has the option of specifying the Y-intercept (A).

Power

The power function is one of the more common functions used in cost analysis work. A plot of its logarithmic counterpart, the log-linear form, is known as the "learning curve" or "improvement cost curve." However, for reasons discussed later, the power, rather than the logarithmic, form is used in this program. For this function,

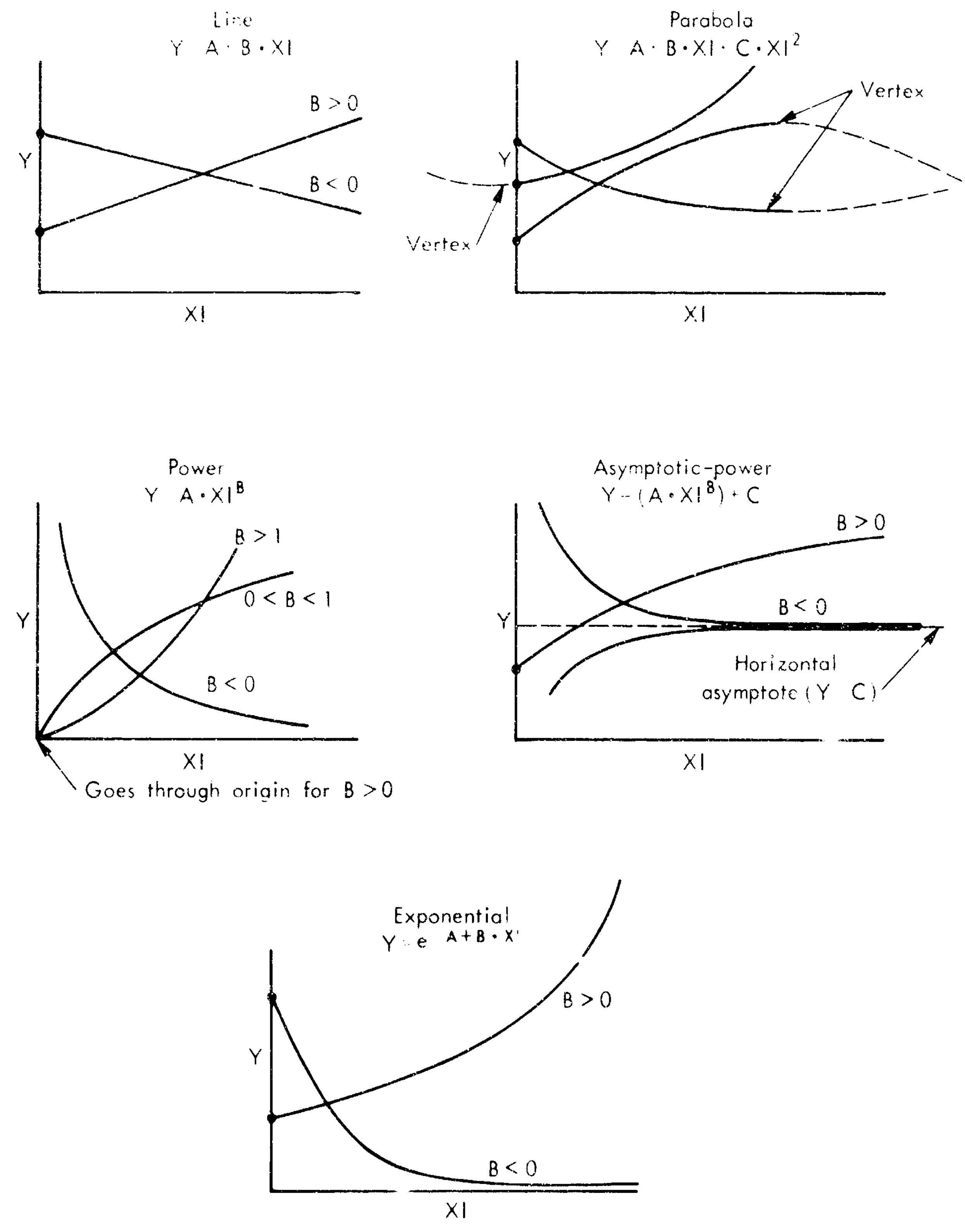


Fig. 1--Examples of Curves used in Program for a One-Independent-Variable Case

the user has the option of using up to three independent variables. For positive exponent B, this curve always passes through the origin, as shown in the figure. Therefore, it should never be used where a positive Y-intercept is desired or logically required. For negative B, the curve is undefined at $X1 = 0$ and is a declining curve, approaching asymptotically the X1-axis as X1 becomes large.

Asymptotic-Power

An examination of the fourth function, the asymptotic-power, shows that the curve has a horizontal asymptote of $Y = C$ for negative B. That is, as X1 becomes large, the first term $(A \cdot X1^B)$ approaches zero, and hence the value of Y approaches that of the constant term C. Consequently, there is a level-off effect associated with this curve for negative B. This function may thus be used to represent points that lie along a curve either increasing or decreasing to a horizontal asymptote. Like the power curve, this curve is undefined at $X1 = 0$ for negative B. For positive B, there is a Y-intercept equal to C. As X1 becomes large, the first term $(A \cdot X1^B)$ ultimately becomes large compared with C, and therefore Y approaches a pure power function $(A \cdot X1^B)$ in this region of X1.

A plot of the asymptotic-power function on log-log paper produces a curved line at low values of X1 that approaches either a horizontal $(B < 0)$ or inclined $(B > 0)$ asymptote at high values of X1. Where a positive Y-intercept is desired, the user may specify a value for the constant term C.

Exponential

The last form, the exponential, is used to represent points that lie along a curve having a positive Y-intercept (e^A). The curve may be either a rising $(B > 0)$ or falling curve $(B < 0)$--the falling curve approaching asymptotically the X1-axis. The logarithmic counterpart of the exponential function is the semilog function, which produces a straight line on semilog paper. That is, ln Y is a linear function of X1.* As was the case with the power function, for reasons discussed later, the exponential, rather than the semilogarithmic, form is retained for this program.

NONLINEAR-LEAST-SQUARES SOLUTIONS

It can be shown mathematically that the least-squares solutions of the parameters of any function are always exact and unique provided that the function is linear with respect to all of its parameters. Therefore, for this program, the line and parabola produce exact and unique solutions. (The term exact is used to refer to a solution that can be obtained algebraically.) However, the latter three functions are not all linear in terms of their parameters. Thus, their solutions are not exact and, as shown later, may not represent absolute minimums of the sum of squares of the Y residuals. They must be obtained in some other way--usually through some type of iterative procedure. (The general principles of such procedures and other mathematical considerations relating to the solutions of nonlinear-least-squares equations are presented in Appendix A.)

For the power and exponential functions, a modified Gauss-Newton method is used, in which initial estimates are obtained from the logarithmic solutions (which are exact) and then corrections, guaranteed to produce convergence to a solution, are applied to those initial estimates. This procedure is repeated until the absolute change in the value of each parameter becomes equal to, or less than, some predetermined value (10^{-8} in the program).**

The solution of the asymptotic-power function is based on another type of iterative procedu because there appears to be no easy way to

---

*All logarithms discussed herein are natural logarithms (base e) and are represented by ln.

**This procedure is described in detail in RM-4879-PR by C. A. Graver and H. E. Boren, Jr., Multivariate Logarithmic and Exponential Regression Models, The RAND Corporation, July 1967. It may be noted that conceptually, the solutions for the power and exponential functions are each different than for their logarithmic counterparts (see Appendix A). Also, the term "exponential" in the above referenced RM is equivalent to the term "power" in this Memorandum.

obtain the initial guesses that are required for the modified Gauss-Newton method. This procedure is treated in Appendix B.

STATISTICAL CONSIDERATIONS

Because of the difficulty of calculating and applying predictive-type statistics for the nonlinear functions, it was decided to use only "goodness of fit" statistics in the program.* Consequently, this program should be regarded as essentially a curve-fitting program with only those statistics being used that relate to how well the curve fits that particular set of data. Also, it should be noted that the statistics may not have exactly the same meaning for the power, asymptotic-power, and exponential functions as for the line and parabola because of the nonlinear characteristics of the former three. In general, statistics for nonlinear functions should be used with care. For example, unless there is proof to the contrary, the F statistic for a nonlinear function probably should not be compared with the F table. Such statistics should generally be used only qualitatively--not quantitatively--until a thorough investigation is made into the application of such statistics to nonlinear functions.

The principal reason for omitting the logarithmic forms in this program is that it is very difficult to compare their fits statistically with those of the nonlogarithmic forms (see Appendix A). As a result, no logarithmic curves are used, and the statistical results relating to the five functions used in this program can be compared more directly. However, since the iterative solutions for the power and exponential functions require that their logarithmic solutions be determined for the initial estimates, these solutions are also printed in the output (without any related statistics) for the benefit of the user.

A summary of the statistical equations is presented in Section III, following the discussion on program outputs.

---

*The use of predictive statistics for the power function is treated in RM-4879 (see previous footnote). However, that program requires many additional subroutines, which in the opinion of the author, would make the CURVES program prohibitively large, slow in operation, and restricted to 50 or less data points.

II. INPUT PROCEDURES

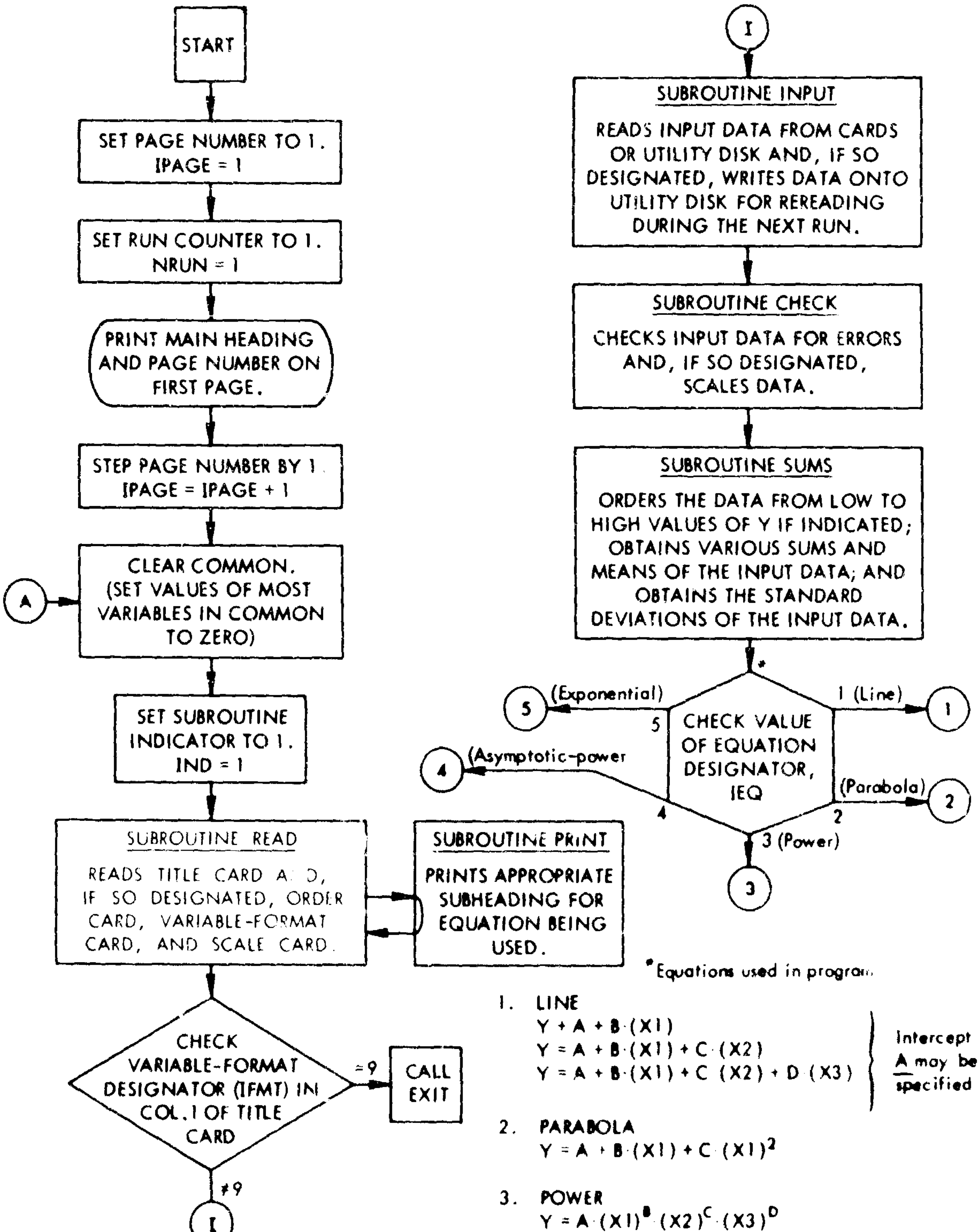
The flow of operations within the program is depicted in Fig. 2. The program is so structured that many sets of data may be entered, in which each set (≤ 200 data points) constitutes a run. As soon as each set is read in, the program operates on that set before proceeding to the next set of input data. Each data set may be entered on a separate deck of cards. On the other hand, several or all of the data sets, if space on the cards permits, may be entered on one deck of cards, thus effecting considerable savings in the use of cards and in the effort of duplicating a deck of cards containing data for several runs. A variable format procedure is used, allowing much flexibility in the format of the input data.

Listed below are the types of cards that must be entered for the first run.

1.	Title card	
2.	Order card	Need to be entered only once if input data for all runs are to be entered in same format
3.	Format card	
4.	Scale card	Used only if data are to be scaled or if Y-intercept is to be specified.
5.	Data cards	
6.	Blank card	
7.	End card	Optional

TITLE CARD

The title card must be entered for each run. In addition to the title (alphanumeric), this card also contains other information about the run. If the two cards relating to the variable-format procedures (order and format cards) are to be read, a "1" is entered in Col. 1 of the title card. For the first run, a "1" must be entered. For subsequent runs involving different input formats, it still must be entered. However, if data for all subsequent runs are to be entered on separate



*Equations used in program

1. LINE
   $Y + A + B \cdot (X1)$
   $Y = A + B \cdot (X1) + C \cdot (X2)$
   $Y = A + B \cdot (X1) + C \cdot (X2) + D \cdot (X3)$
   Intercept A may be specified
2. PARABOLA
   $Y = A + B \cdot (X1) + C \cdot (X1)^2$
3. POWER
   $Y = A \cdot (X1)^B \cdot (X2)^C \cdot (X3)^D$
4. ASYMPTOTIC-POWER
   $Y = [A \cdot (X1)^B] + C$
   Intercept C may be specified
5. EXPONENTIAL
   $Y = e^{[A+B \cdot (X1)]}$

Fig. 2--Flow of Operations

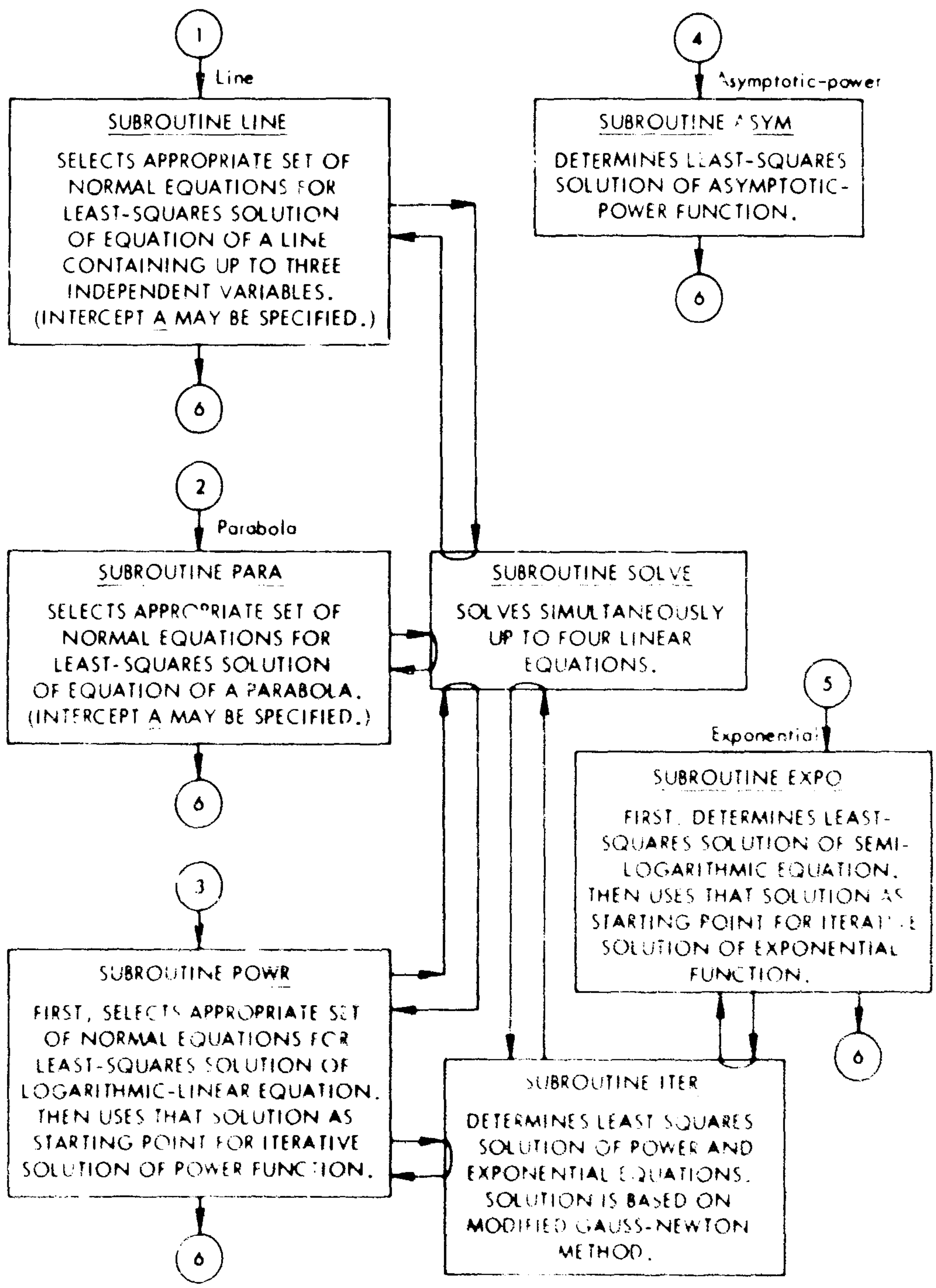


Fig. 2--Flow of Operations (Cont.)

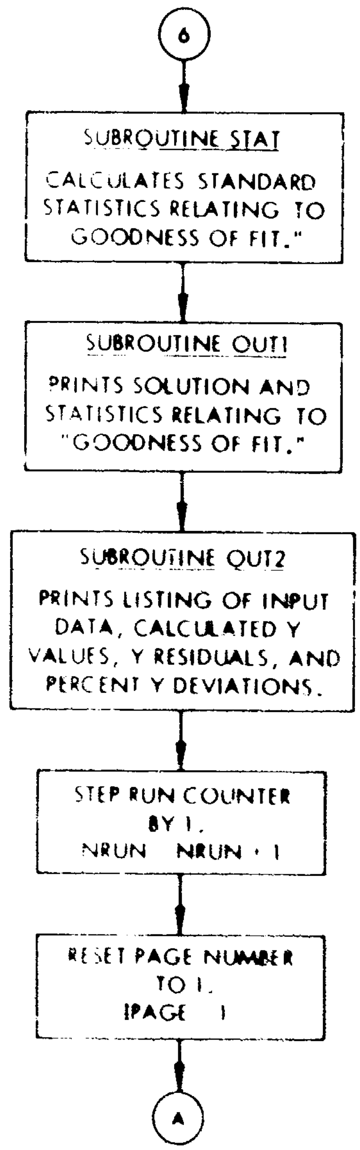


Fig. 2--Flow of Operations (Cont.)

decks of cards using the same format as the first run, then the "1" only needs to be entered for the first run. Whenever the 1 is not entered, then the order and format cards are not entered.

If input data representing several runs are entered on one deck of cards so that for subsequent runs those same cards will, in effect, be reused, and hence re-read (in different fields), then a "1" is entered in Col. 2 of the title card for the first set of such data to be read in. This causes the machine not only to read the first set of data but to write all of the input data onto a utility disk for rereading during subsequent runs. Unless the user selects another utility disk or tape, the program automatically uses utility disk S.SU04 (FORTRAN logical unit 4) for this operation.

For the remaining runs that use the input data from the same deck of cards, a "2" is entered in Col. 2 of each title card for those runs. This causes the machine to read the input data from the utility disk (instead of from new cards) in accordance with the format instructions so entered.

Column 3 is used for the function designator. An integer from "1" through "5" is entered to designate which function is being considered for that run. The integer designators are as follows:

Integer	Function
1	Line
2	Parabola
3	Power
4	Asymptotic-power
5	Exponential

There must be an integer of one of the above values entered in Col. 3 for the first run. If Col. 3 is left blank after the first run, then the value for the previous run is used. Thus, if the same type of function is being examined for a series of runs, its designator needs to be entered only for the first run.

Column 4 is used to designate whether the input data are to be ordered from low to high values of Y. A value of "1" signifies that the data are to be ordered. For the first run a blank (zero) signifies that the data are not to be ordered. However, for subsequent runs a blank (zero) signifies to the machine that the value of the order designator for the preceding run is to be used. Again, this is done so that if all runs in a series are to be either ordered or unordered, the order designator will only have to be entered for the first run. In the case where a zero is desired for the designator after a "1" has been entered previously, a "2" must be entered. This in effect sets the value of the designator to zero.

The scale designator is entered in Col. 5. If any of the data are to be scaled (using the Scale Card described later), a "1" is entered in this column. Otherwise it is left blank (zero).

Column 6 is used to designate whether the Y-intercept term for the line, parabola, or asymptotic-power function is to be specified. This is done by entering a "1." Otherwise Col. 1 is left blank (zero). If a "1" is entered in either Cols. 5 or 6, or both, then the scale card is entered in the order shown previously.

Columns 9 through 72 are reserved for the title. The title may consist of any alphanumeric symbols.

A summary of the information on the title card is given in Table 1. An example of a title card is shown in Fig. 3, in which a linear regression is to be made on the input data ("1" in Col. 3). The data are to be ordered with respect to Y ("1" in Col. 4) and are to be scaled ("1" in Col. 5). The "1" in Col. 1 indicates that the order and format cards are to be read next.

ORDER CARD

The next card (when used as specified by a "1" in Col. 1 of the title card) indicates the order (from left to right) in which the data are located on the data cards. The order (designated by alphanumeric symbols) is entered in Cols. 1-2, 4-5, 7-8, 10-11, and 13-14. Depending on the number of independent variables being used and on whether identifiers are being used, Cols. 7-14 may not be required. The symbols used to show the order are as follows:

Table 1

SUMMARY OF INFORMATION ON TITLE CARD

Columns	Remarks
1	A "1" indicates that the order card and format card are to be read, respectively, following the title card If blank, no such cards are to be read. A "1" must be entered for the first run.
2	A "1" indicates that the input data cards contain data for several runs and are to be written onto a utility disk (System Unit S.SU04, FORTRAN Logical Unit 4). A "2" indicates that the input data are to be read from the disk. If blank, the input data cards for this run are to be read only once.
3	An integer from "1" to "5" is used to designate which function is being used. This is done as follows. 1 -- Linear 2 -- Parabolic 3 -- Power 4 -- Asymptotic-power 5 -- Exponential If blank after the first run, the value for the preceding run is used.
4	A "1" indicates that the data are to be ordered from low to high values of Y. For the first run a blank indicates that the data are not to be ordered. If blank for subsequent runs, the value for the preceding run is used. A "2" is used for subsequent runs to restore the designator to zero when desired.
5	A "1" indicates that the data are to be scaled. Otherwise, it is left blank.
6	A "1" indicates that a Y-intercept is to be specified for either the linear, parabolic, or asymptotic-power case. Otherwise, it is left blank.

Table 1 (Cont.)

SUMMARY OF INFORMATION ON TITLE CARD

Columns	Remarks
7-8	Not used.
9-72	Title for run. May consist of any alphanumeric symbols.

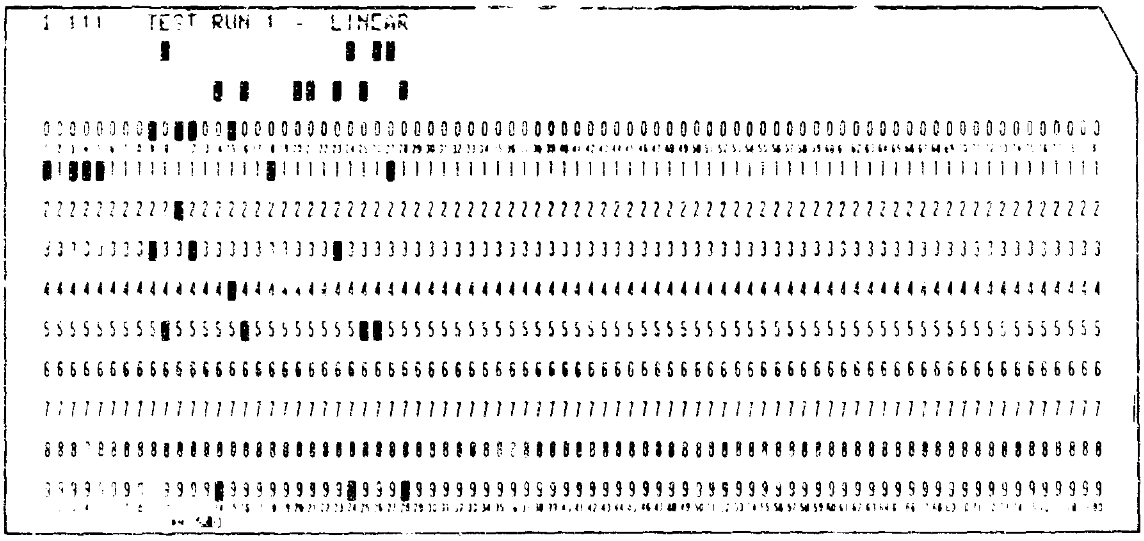


Fig. 3--Example of Title Card

Symbol	Type of Data	
ID	...... Identifier (alphanumeric) (optional)	May be in any order from left to right.
Y1	...... Dependent variable (required)	
X1	...... First independent variable (required)	
X2	...... Second independent variable (optional)	
X3	...... Third independent variable (optional)	

Suppose that a set of data is to be entered in which values for the three independent variables and the dependent variable are located in Cols. 1-12, 13-24, 25-36, and 37-48, respectively. Suppose also that an identifier (a six-digit integer) is in Cols. 55-60. Then, "X1," "X2," "X3," "Y1," and "ID" would be entered respectively, in Cols. 1-2, 4-5, 7-8, 10-11, and 13-14 to show the above order across the card.

All alphanumeric information is treated in A4 formats in this program in order to be adaptable to IBM-360 systems. In addition, the identifiers may be entered in either A4 or 2A4 formats. This is indicated in Col. 16 on the order card by either a "1" (A4) or a "2" (2A4). If Col. 16 is left blank, after the first run, then the value for the preceding run is used.

The order card is shown in Fig. 4 for the above example, in which a 2A4 format is to be read for the six-digit identifier. Since the commas separating the symbols are in columns that are not read by the machine, they may be used for the purpose of clarification.

FORMAT CARD

The format card indicates where the data are located on the data cards. Again, this card is used only if a "1" is entered in Col. 1 of the title card. This card must begin with a left-hand parenthesis and end with a right-hand parenthesis before , and the information within the parentheses must conform to the rules for FORTRAN formats. Except for the alphanumeric identifiers, all input data must be in real-number (floating point) formats. The format card shown in Fig. 5 would be used for the previous example.

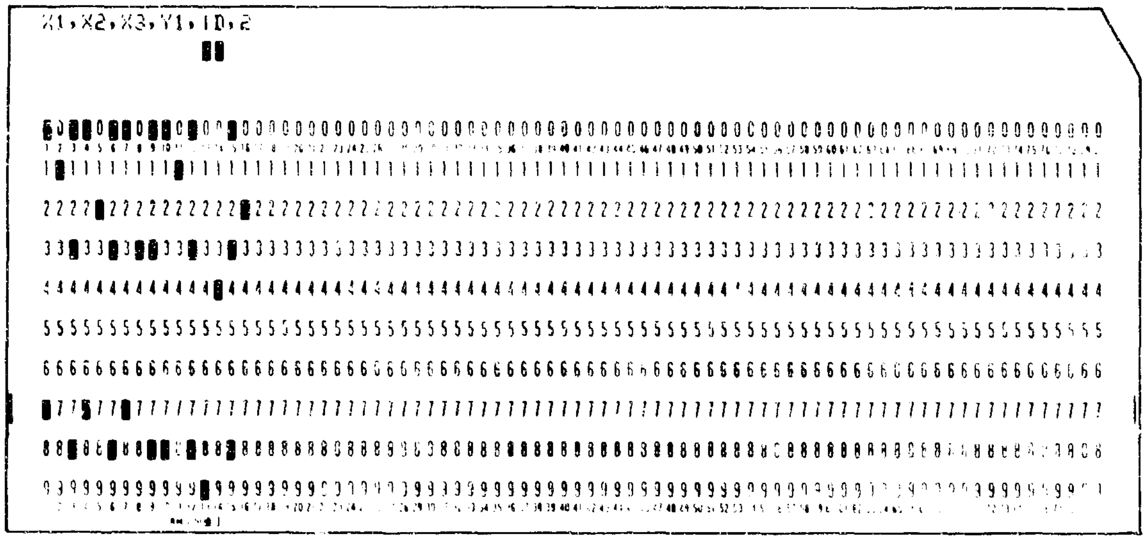


Fig. 4--Example of Order Card

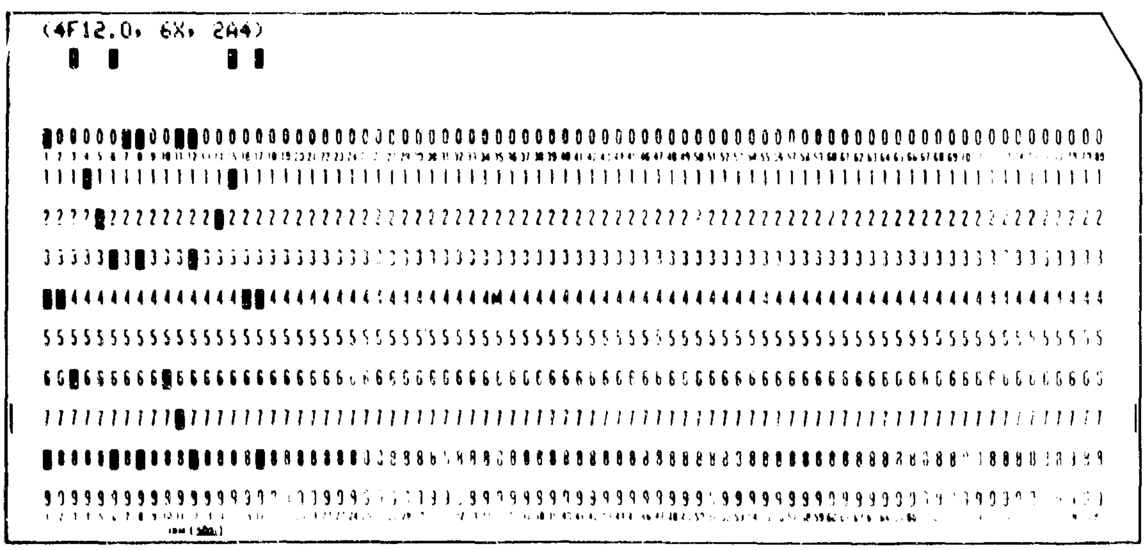


Fig. 5--Example of Format Card

SCALE CARD

As designated by a "1" in either Cols. 5 or 6 (or both) of the title card, the scale card is used either if the data are to be scaled, or if the Y-intercept is to be specified for the linear, parabolic, or asymptotic-power function.

Data Scaling

The first four sets of two columns each on the scale card are used for scaling the input dependent and independent variable data when required.* These scale indicators (integers) are located as follows:

Variable Scaled	Column Location of Scale Designator
Y (dependent variable)	1-2
X1 (first independent variable)	3-4
X2 (second independent variable)	5-6
X3 (third independent variable)	7-8

An integer (fixed-point) number is used to indicate the number of places that the decimal is to be moved. A positive number indicates that the decimal is to be moved to the right as many places as is the value of the number. A negative number indicates that the decimal is to be moved to the left as many places as is the absolute value of the number. For example, suppose that a "3" is entered in Col. 2 (must be right-justified). Then each input Y value will have its decimal point moved three places to the right, e.g., 50.123 ⟶ 50123., 0.6127 ⟶ 612.7, etc.

Care must be taken to enter all positive integers in the right-hand column of the two-column set. If, for example, a "3" were entered in Col. 1 instead of Col. 2, the machine would read the number as "30" instead of "3." Any scale factor entered applies, of course, to the entire set of the corresponding variable for that run only. An example of this card is shown in Fig. 6, in which the first-independent-variable data (X1) are to be scaled down by a factor of 10.

---

*Data may also be scaled by using P-formats.

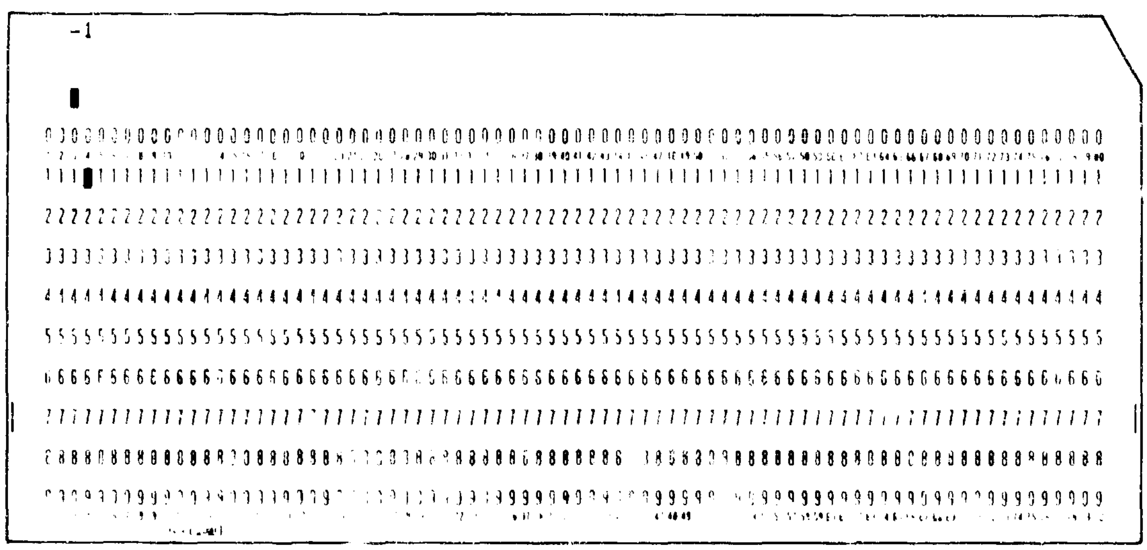


Fig. 6--Example of Scale Card

Specified Y-Intercept

The specified Y-intercept (regression constant) is entered in Cols. 11-20. It is entered in real-number form anywhere within these columns. The implied decimal point location is at the right end of the field, between Cols. 20 and 21.

DATA CARDS

Each data card must contain at least a pair of values--one for the dependent variable (Y) and one for the independent variable (X1). Each data set constituting a run must contain one more data point than the number of parameters being solved and may contain up to 200 data points. The location of the data on the cards must be in exact agreement with the information entered on the order and format cards, or else the data will not be read properly. The numerical data (dependent and independent variable values) must be entered as real numbers, and the identifiers (if used) as alphanumeric data.

If, for any data card, either the Y field or any of the X fields, that are read, but not all, are blank or contain zeros, that card is skipped. However, if all X-Y fields are blank (zeros), the reading of input data for the run is terminated at that point (see Blank Card below).

In conjunction with the above, zero or negative values are not allowed for the X-Y input data. Another reason for this is that for the power and asymptotic-power functions involving negative fractional exponents of negative X values, such expressions are meaningless. Also, logarithms are used in the solutions of the nonlinear functions, and the use of logarithms restrict the input data to values greater than zero.

An example of a data card containing data in the format depicted in Figs. 4 and 5 is shown in Fig. 7.

BLANK CARD

Each pack of data cards must always end with a blank card. This card is used to terminate the reading of the input data for a given run.

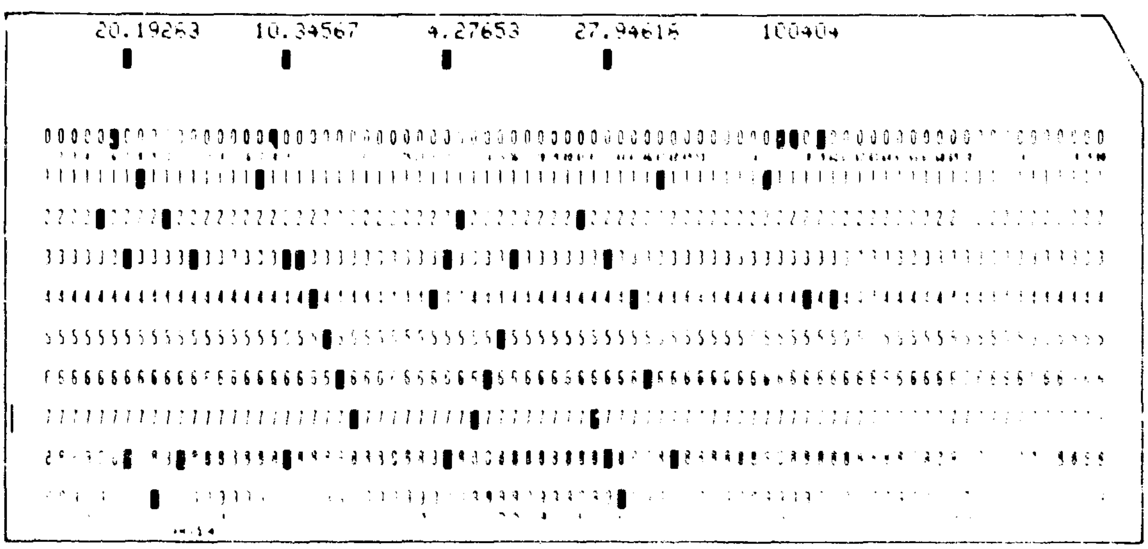


Fig. 7--Example of Data Card

END CARD (OPTIONAL)

The program is so structured that after each set of data is read in and processed, the machine attempts to read in another set of data. After the final run when the machine attempts to read in another set of data that does not exist, control is returned to the FORTRAN monitor, ending the series of runs. At this point, a statement is printed as follows:

End-of-data encountered on system input file.

In order to provide a positive method of terminating a series of runs, this optional method is provided the user. If a "9" is entered in Col. 1 on a card immediately following the blank card of the last set of data (actually the next title card), it will cause the program to Call Exit and terminate the series of runs at that point. If this is done, the End-of-Data statement will not be printed.

SUMMARY

Figure 8 shows the order of two data decks for a series of three runs. In this figure, the second deck of data also contains data for the third run. Hence, it is to be written onto a utility disk for re-reading during the third run.

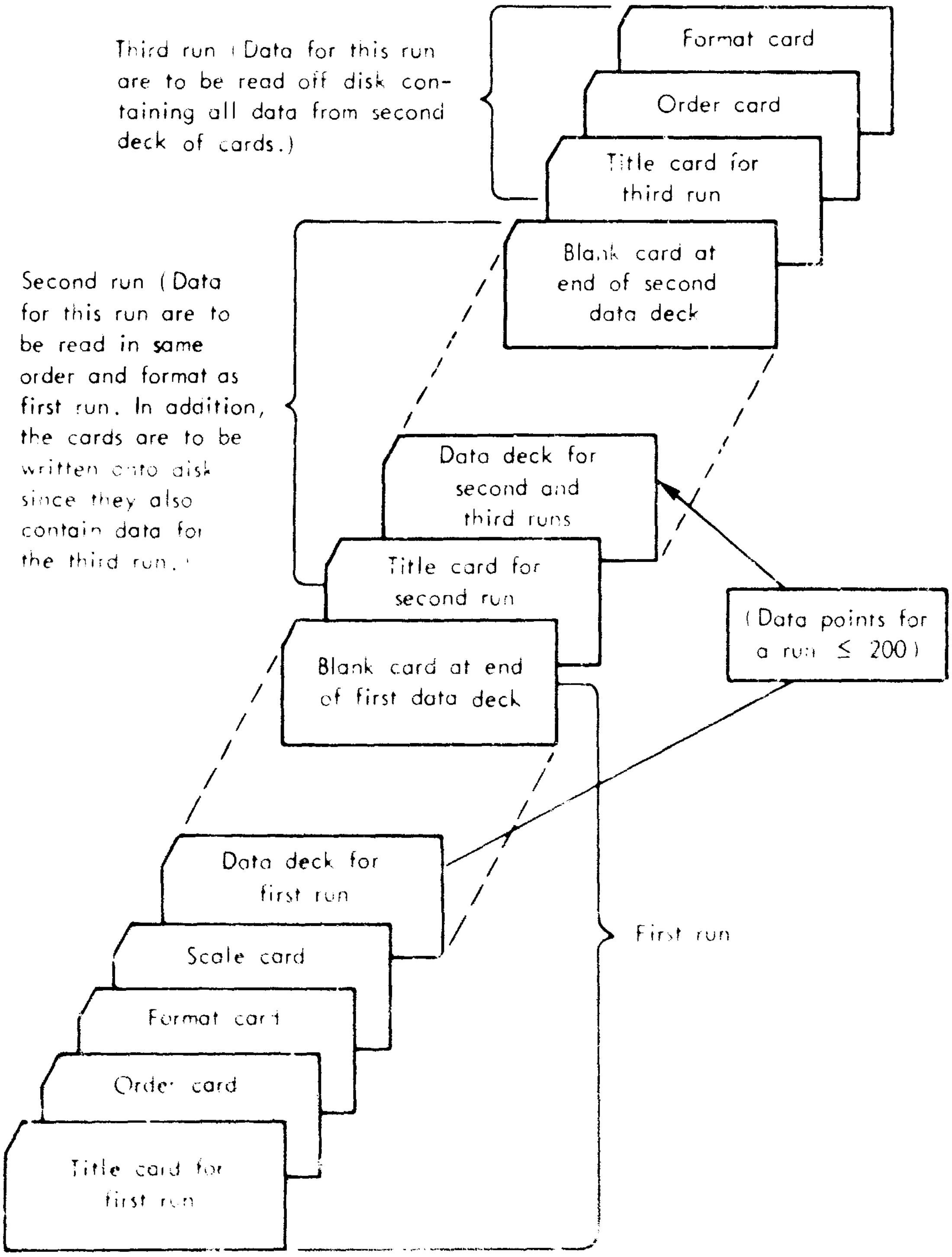


Fig. 8--Arrangement of Two Data Card Decks for Three Runs

III. PROGRAM OUTPUTS

Examples of program outputs for the line and power functions are shown in Figs. 9-12. The particular data shown have no meaning whatsoever and are presented only for the purpose of displaying the outputs. Since the headings are rather self-explanatory, and the statistics have already been discussed in Chapter I, no explanation of them will be given. It should be emphasized again that the statistics presented may have different meanings with regard to the nonlinear functions than with the linear functions. They should not be used as predictors and should be regarded only as indicators of the "goodness of fit" for that particular data set.

As is shown in Appendix A for the power and exponential functions, if there is a solution of the parameters lying in the region defined by

$$2 \cdot Y_{ci} - Y_i > 0,$$

where Y_{ci} = computed value of Y at each point i,

Y_i = observed value of Y at each point i,

then that solution represents an absolute minimum of the sum of squares of Y residuals and is the only solution in that region. In addition, it seems reasonable to assume that the above condition should hold for those functions in order to have a "good" fit. If the above condition for the region does not hold, a message is printed noting that fact.

For the parabolic function, the X-Y coordinates of the maximum or minimum point (vertex) are printed. After this point is passed, the effect of X1 on Y is reversed.

Figure 13 is a listing of the input data for the two runs whose outputs are shown in Figs. 9-12. The data are in the same format on the cards as shown in Figs. 4-7. As Fig. 13 shows, the input data for the second run (power function) are read in the same format as for the first run. Therefore, the order and format cards are not

```
                         CURVES REGR SSION ANALYSIS COMPUTER PROGRAM                      PAGE  1

TEST RUN 1 -- LINEAR

               LINEAR REGRESSION -- Y = A + (B * X1) + (C * X2) + (D * X3)
X1 SCALE FACTOR      0.1E 00

                                   SUMMARY TABLE

     A                                                                  10.88890
     B                                                                  11.59367
     C                                                                  -0.76324
     D                                                                   0.64312

     COEFFICIENT OF CORRELATION (UNADJUSTED)                             0.94126
     COEFFICIENT OF DETERMINATION (UNADJUSTED)                           0.88598
     STANDARD ERROR OF THE ESTIMATE OF Y (ADJUSTED)                     14.19868
     COEFFICIENT OF VARIATION (PERCENT)                                 17.50770
     SUM OF SQUARES OF Y RESIDUALS                                    5212.17694
     MEAN OF ABSOLUTE PERCENT Y DEVIATIONS                              10.98372

     F VALUE                                                            67.34206
     DEGREES OF FREEDOM ABOUT REGRESSION CURVE                          26.00000
     DEGREES OF FREEDOM DUE TO REGRESSION                                3.00000
     TOTAL DEGREES OF FREEDOM                                           29.00000

     MEANS OF INPUT DATA
       Y                                                                80.87119
       X1                                                                6.32042
       X2                                                               45.39382
       X3                                                               48.75018

     STANDARD DEVIATIONS OF INPUT DATA
       Y                                                                39.70236
       X1                                                                3.33733
       X2                                                               41.98536
       X3                                                               36.82856

     NUMBER OF DATA POINTS                                              30.
```

Fig. 9--First Page of Output (Linear Case)

TEST RUN 1 -- LINEAR

COMPUTED Y VALUES AND RESIDUALS

LABEL	Y	X1	X2	X3	Y CALC.	Y DEV.	PERCENT Y DEV.
100101	14.39608	5.12227	86.12357	24.16273	20.08085	-5.68477	-39.48829
100202	19.44080	0.23766	15.28765	30.98716	21.90433	-2.46353	-12.67196
100303	24.06789	1.52677	40.15672	63.17772	38.57032	-14.50243	-60.25636
100404	27.94616	2.01928	10.34567	4.27653	29.15382	-1.20764	-4.32131
100505	33.60992	3.52567	42.16543	27.17864	37.06082	-3.45090	-10.26751
100606	40.40568	3.82761	27.37677	3.28716	36.48385	3.92183	9.70615
100707	40.71304	1.21549	3.26751	31.26884	42.59641	-1.88337	-4.62597
100808	40.97917	12.11762	175.26876	57.17675	51.15903	-10.17986	-24.84155
100909	51.71629	3.22761	8.18761	16.27615	52.52709	-0.81080	-1.56778
101010	54.53670	8.92672	109.26547	44.27861	59.46228	-4.92558	-9.03167
101111	59.00000	1.00000	20.00000	101.00000	72.17233	-13.17233	-22.32598
101212	71.09687	4.01987	15.00000	39.22218	71.26912	-0.17225	-0.24227
101313	80.09382	5.03729	10.00000	20.00000	74.51941	5.57441	6.95985
101414	83.40204	8.02456	35.25411	15.25672	86.82731	-3.42527	-4.10694
101515	84.84002	6.53382	16.27865	1.27553	75.03557	9.80445	11.55640
101616	88.26627	5.62718	19.26713	41.26517	87.96131	0.30496	0.34550
101717	89.92674	6.21786	18.88888	29.17656	87.32377	2.60297	2.89455
101818	90.28862	7.51234	76.11111	88.99112	97.12469	-6.83607	-7.57136
101919	92.51133	6.22452	10.27625	17.24561	86.30151	6.20982	6.71249
102020	94.38172	4.92882	14.11167	54.28817	92.17486	2.20686	2.33823
102121	100.47238	10.53547	55.27618	12.16547	98.66815	1.80423	1.79574
102222	104.27262	9.42469	48.23418	35.28861	106.03578	-1.76316	-1.69091
102323	107.75168	10.00000	100.00000	100.00000	114.81265	-7.06097	-6.55300
102424	111.35247	9.72617	84.23456	93.25671	119.33416	-7.98169	-7.16795
102525	122.69347	8.92219	12.16524	8.[illegible]187	110.19726	12.49621	10.18490
102626	131.24108	11.73392	95.18293	100.00000	138.59176	-7.35069	-5.60090
102727	137.00000	10.62679	81.27543	107.26784	141.04492	-4.04492	-2.95249
102828	138.46147	7.42819	18.22184	97.22215	145.62622	-7.16475	-5.17454
102929	143.72068	6.41625	102.33728	114.23477	80.63456	63.08612	43.89495
103030	147.55076	7.92619	12.25418	90.26713	151.48166	-3.93089	-2.66410

Fig. 10--Second Page of Output (Linear Case)

```
TEST RUN 2 -- POWER                                                          PAGE  1

                POWER REGRESSION -- Y = A * (X1**B) * (X2**C) * (X3**D)

                                   SUMMARY TABLE

A                                                              15.11007
B                                                               1.18696
C                                                               0.27451
D                                                              -0.55490

LOGARITHMIC SOLUTIONS
  A                                                            13.27378
  B                                                             1.20099
  C                                                             0.29947
  D                                                            -0.56444

COEFFICIENT OF CORRELATION (UNADJUSTED)                         0.99824
COEFFICIENT OF DETERMINATION (UNADJUSTED)                       0.99649
STANDARD ERROR OF THE ESTIMATE OF Y (ADJUSTED)                 33.08461
COEFFICIENT OF VARIATION (PERCENT)                              4.27699
SUM OF SQUARES OF Y RESIDUALS                               28459.37622
MEAN OF ABSOLUTE PERCENT Y DEVIATIONS                           3.99534

F VALUE                                                      2432.77188
DEGREES OF FREEDOM ABOUT REGRESSION CURVE                      26.00000
DEGREES OF FREEDOM DUE TO REGRESSION                            3.00000
TOTAL DEGREES OF FREEDOM                                       29.00000

MEANS OF INPUT DATA
  Y                                                           773.54961
  X1                                                           65.46327
  X2                                                           65.01890
  X3                                                           65.94354

STANDARD DEVIATIONS OF INPUT DATA
  Y                                                           528.96464
  X1                                                           39.00866
  X2                                                           40.25761
  X3                                                           48.50177

NUMBER OF DATA POINTS                                          30.
```

Fig. 11--First Page of Output (Lower Case)

TEST RUN 2 — POWER

PAGE 2

COMPUTED Y VALUES AND RESIDUALS

LABEL	Y	X1	X2	X3	Y CALC.	Y DEV.	PERCENT Y DEV.
200101	60.25169	10.21822	65.27819	95.11118	59.95306	0.29863	0.49564
200202	115.19201	29.12678	8.16279	75.25619	133.76512	-18.57311	-16.12361
200303	143.04087	25.14567	34.25671	90.00000	150.82014	-7.77927	-5.43850
200404	247.96037	38.29918	27.18827	75.27164	257.55107	-9.59070	-3.86784
200505	342.15482	35.12311	19.23519	37.26153	312.19974	29.95508	8.75483
200606	351.20949	35.18762	10.26781	16.25673	417.28365	-66.07416	-18.81332
200707	370.88634	17.26155	76.25144	10.15782	403.32697	-32.44062	-8.74678
200808	425.47155	41.15287	118.26132	77.23518	413.96928	11.50227	2.70342
200909	460.16664	20.23457	14.16289	4.25617	497.17656	-37.00991	-8.04272
201010	461.97847	23.15678	104.28715	19.29175	436.37669	25.60178	5.54177
201111	530.19078	67.22218	48.18273	92.23116	524.94786	5.24292	0.98887
201212	531.15436	71.14253	100.18273	145.27168	533.65966	-2.50530	-0.47167
201313	535.55565	70.27168	108.26152	132.17817	565.95119	-30.39555	-5.67552
201414	585.28353	89.28615	5.27715	45.23519	594.75689	-9.47336	-1.61859
201515	661.12500	74.28719	66.26718	100.00000	556.40512	4.71988	0.71392
201616	755.79487	44.27651	85.28716	27.18279	736.83550	18.95937	2.50853
201717	779.62800	28.12816	9.18826	3.27715	754.58806	25.03994	3.21178
201818	800.00000	43.13425	59.17236	18.21926	806.74354	-6.74354	-0.84294
201919	814.87900	111.25411	23.18726	76.16233	868.48782	-53.60883	-6.57875
202020	880.20145	102.17628	112.27162	130.18719	898.95210	-18.75065	-2.13027
202121	972.57176	34.18273	26.18273	5.26174	974.85854	-2.28677	-0.23513
202222	980.76959	100.00000	100.00000	100.00000	982.68075	-1.91125	-0.19487
202323	1000.58891	121.27157	55.24351	110.23145	994.50979	6.07912	0.60755
202424	1125.67638	151.28761	93.27615	178.29977	1143.32674	-17.65036	-1.56798
202525	1138.31287	120.11234	77.19283	105.25172	1105.81754	32.49533	2.85469
202626	1158.45163	120.17865	90.24133	105.25671	1154.98471	3.46692	0.29927
202727	1345.54034	57.19287	102.00000	17.26153	1349.24675	-3.70641	-0.27546
202828	1472.22902	65.24138	110.27816	21.25517	1435.85831	36.37071	2.47045
202929	1510.21359	102.15672	54.28716	40.25617	1412.08621	98.12738	6.49758
203030	2650.01077	112.18882	147.23557	25.18892	2692.09235	-42.08157	-1.58799

Fig. 12--Second Page of Output (Power Case)

```
1 111    TEST RUN 1 -- LINEAR                                                      Title card
X1,X2,X3,Y1,10,2                                                                   Order card
(4F12.0, 6X, 2A4)                                                                  Format card
  -1                                                                               Scale card
    20.19283      10.34567       4.27653      27.94618         100404
    50.37289      10.0          20.0          80.09382         101313
   105.35467      55.27618      12.16547     100.47238         102121
    62.17862      18.88888      29.17654      89.92674         101717
    35.25671      42.16543      27.17864      33.60992         100505
    80.24561      35.25411      15.25672      83.40204         101414
    12.15487       3.26751      31.26884      40.71304         100707
    65.33821      16.27865       1.27553      84.84002         101515
    89.26718     109.26547      44.27861      54.53670         101010
   100.0         100.0         100.0         107.75168         102323
   121.17625     175.26876      52.17625      40.97917         100808
     2.37658      15.28765      30.98716      19.44080         100202
   106.26789      81.27543     107.26784     137.0             102727
    56.27182      19.26713      41.26517      88.26627         101616
    15.26718      40.15672      63.17772      24.06789         100303
    79.26182      12.25418      30.26713     147.55076         103030   Input data
    51.22268      86.12357      24.16273      14.39608         100101
    97.26173      84.23456      93.25671     111.35247         102424
    62.24518      10.27625      17.24561      92.51133         101919
    32.27615       8.18761      16.27615      51.71629         100909
    38.27615      27.37677       3.28716      40.40568         100606
    49.28817      14.11167      54.28817      94.38172         102020
    94.24689      48.23418      35.28861     104.27262         102222
    40.19824      15.0          39.22218      71.09687         101212
   117.33922      95.18293     100.0         131.24108         102626
    75.12345      76.11111      88.99112      90.28862         101818
    10.0          20.0         101.0          59.0             101111
    74.28192      18.22184      97.22215     138.46147         102828
    89.22186      12.16524       8.01187     122.69347         102525
    64.16253     102.33728     114.23477     143.72068         102929
                                                                                   Blank card
  3      TEST RUN 2 -- POWER
    35.18762      10.26781      16.25673     351.20949         200606
   102.15672      54.28716      40.25617    1510.21359         202929
    78.28719      66.26718     100.0         661.12500         201515
   102.17628     112.27162     130.18719     880.20145         202020
    70.27168     108.26152     132.17817     535.55565         201313
    43.13425      59.17236      18.21926     800.0             201818
    10.21822      65.27819      95.11118      60.25169         200101
    67.22218      48.18273      92.23116     530.19078         201111
   100.0         100.0         100.0         980.76950         202222
   120.11234      77.19283     105.25172    1138.31287         202525
    34.18273      26.18273       5.26174     972.57176         202121
    29.12678       8.16279      75.25619     115.19201         200202
    20.23457      14.16289       4.25617     460.16664         200909
    57.19287     102.0          17.26153    1345.54035         202727
   120.17865      90.24133     105.25671    1158.45163         202626
    25.14567      34.25671      90.0         143.04087         200303
    17.26155      76.25144      10.15782     370.88634         200707
    65.24138     110.27816      21.25517    1472.22902         202828
    28.12816       9.18826       3.27715     779.62800         201717
    41.15287     118.26132      77.23518     425.47155         200808
    35.12311      19.23518      37.26153     342.15482         200505
   121.27157      55.24351     110.23145    1000.58891         202323
    23.15678     104.28715      19.29175     461.97847         201010
   111.25411      23.18726      76.16233     814.87900         201919
   151.28761      93.27615     178.29977    1125.67637         202424
    44.27651      85.28716      27.18279     155.79487         201616
    71.16253     100.18273     145.27168     531.15436         201212
   112.18882     147.23557      25.18892    2650.01076         203030
    38.29918      27.18827      75.27164     247.96037         200404
    89.26615       5.27715      45.23519     585.28353         201414
                                                                                   Blank card
```

Fig. 13--Listing of Inputs for Two Runs whose Outputs are Shown in Figs. 9-12

required for that second run. Also, there is no scale card for the second run because none of the data are scaled.

A summary of all of the statistics that are used in the CURVES program is presented in Table 2 in simplified form. One should be aware that when the Y-intercept is specified, not only are the degrees of freedom changed, but the curve is no longer forced through the means of the observed data. Consequently, the statistics will have different values. In addition, for the linear case with the specified intercept, the F ratio, which is based on the distribution of the Y residuals, may have its distribution altered.* Therefore, a comparison of its value with tabulated values based on an assumed standard F distribution may not always be valid. As a result, F ratios for the specified Y-intercept cases may not always be comparable with those for unspecified Y-intercept cases. A statement to this effect is printed in the Summary Table of the output whenever the Y-intercept is specified.

A listing of the FORTRAN-IV computer program is presented in Appendix C.

---

*As was noted previously in Section I under <u>Statistical Considerations</u>, F ratios for nonlinear functions may not be valid for comparisons with tabulated values.

Table 2

STATISTICAL EQUATIONS USED IN PROGRAM[a]

Statistic	Equation
Sum of Squares of Y Deviations	$$S_1 = \sum_{i=1}^{N} (Y_i - Y_{ci})^2$$ where N = number of data points Y_i = observed Y value at each point i Y_{ci} = calculated Y value at each point i
Standard Error of the Estimate of Y (Adjusted)	$$S_2 = \sqrt{\frac{S_1}{N-P}}$$ where P = number of parameters to be solved
Coefficient of Variation (Percent)	$$S_3 = \left(\frac{S_2}{\overline{Y}}\right) \cdot 100$$ where $\overline{Y}$ = mean of observed Y values
Coefficient of Determination (Unadjusted)	$$S_4 = 1 - \frac{S_1}{\sum_{i=1}^{N} (Y_i - \overline{Y})^2}$$
Coefficient of Correlation (Unadjusted)	$$S_5 = \sqrt{S_4}$$
Percent Y Deviation	$$S_{6_i} = 100 \cdot \left(\frac{Y_i - Y_{ci}}{Y_i}\right)$$

[a]The symbols used here are not necessarily those used in the program.

Table 2 (Cont.)

STATISTICAL EQUATIONS USED IN PROGRAM

Statistic	Equation
Mean of Absolute Percent Y Deviations	$S_7 = \dfrac{\sum_{i=1}^{N} \lvert S_{6_i} \rvert}{N}$
Standard Deviation of Input Variables	$S_8 = \sqrt{\dfrac{\sum_{i=1}^{N} (V_i - \bar{V})^2}{N-1}}$ where V_i = value of any input variable $\bar{V}$ = mean value of input variable V
F Value	$S_9 = \dfrac{\left[\sum_{i=1}^{N} (Y_{ci} - \bar{Y}_c)^2\right] / (P-1)}{S_1 / (N-P)}$ where $\bar{Y}_c$ = mean of calculated Y values
Total Degrees of Freedom	$S_{10} = N-1$
Degrees of Freedom About Regression Curve	$S_{11} = N-P$
Degrees of Freedom Due to Regression	$S_{12} = S_{10} - S_{11} = P-1$

Appendix A

NONLINEAR-LEAST-SQUARES CONSIDERATIONS

LOGARITHMIC AND NONLOGARITHMIC SOLUTIONS

The usual procedure for making least-squares determinations of the parameters of the power or exponential function is to first convert the function into a log-linear (or semilog-linear) function. One then has a logarithmic linear equation which can be solved directly through use of the standard normal least-squares equations for the linear case. However, one should be aware that such a solution is not the same conceptually as the least-squares solution of the function before it is converted into its logarithmic counterpart. This may be seen by considering, for example, the power function and its logarithmic form.

Let

$$Y = A \cdot (X1)^B \cdot (X2)^C \cdot (X3)^D,$$

and

$$\ln Y = \ln A + B \cdot \ln X1 + C \cdot \ln X2 + D \cdot \ln X3$$

For a least-squares solution, one is interested in minimizing the sum of squares of the Y residuals (denoted by Q).* Therefore, for the power function,

$$Q = \sum_{i=1}^{N} (Y_i - Y_{ci})^2 = \min.,$$

and for the logarithmic function

$$Q' = \sum_{i=1}^{N} (\ln Y_i - \ln Y_{ci})^2 = \min.,$$

*Throughout this discussion, Q is used to represent the sum of squares of the Y residuals.

or

$$Q' = \sum_{i=1}^{N} (\ln \frac{Y_i}{Y_{ci}})^2 = \min.,$$

where N = number of data points,

Y_i = observed value of dependent variable at each point i,

Y_{ci} = calculated value of dependent variable at each point i.

In the logarithmic case, the sum of squares of the actual differences (residuals) between the observed and calculated Y values is not being minimized--rather the sum of squares of the logarithms of the ratios of those values is being minimized. Depending on the observed data, Q and Q' may produce significantly different solutions for the parameters A, B, C, D.

It may also be seen that any statistic based on the sum of squares of Y residuals, such as the coefficient of correlation, may be misleading if used to compare the logarithmic form with its nonlogarithmic counterpart. For the logarithmic form, such statistics are in logarithms and hence have different meanings.

The question as to whether, say, the power function or its logarithmic form is more appropriate for a set of data is beyond the scope of this Memorandum. An answer to such a question depends on many factors including the errors associated with the data and what criterion is used for a "good fit." For the interested reader, this question is treated in more detail in RM-4879-PR.*

NON-LINEAR SOLUTIONS

It is a necessary condition that the first partial derivatives of Q with respect to the parameters must be zero in order that Q be minimized. This is not, unfortunately, a sufficient condition for a function that is not linear with respect to all of its parameters. The reason for this is that if Q could be graphed (in multi-dimensional space) for this type of function, there might be other critical

---

*Graver, Boren, op. cit.

points--such as saddle points or relative maxima or minima points--where the first partial derivatives would also be zero. A test that checks for this possibility is the examination of the matrix of second partial derivatives of Q, which is a generalization of the second-derivative test for a one-parameter case. If this matrix is positive-definite for all parameters in a region containing a solution, it can be shown that the solution represents an absolute minimum for Q in that region and is the only solution in that region.* However, if the matrix is not positive-definite in that region, then there may be other "solutions" for the same set of data.

With regard to the power and exponential functions, it can be shown that the matrix of second-partial derivatives of Q will be positive-definite for values of the parameters which lie in the region defined by**

$$2 \cdot Y_{ci} - Y_i > 0, \qquad \text{(coefficient } A > 0 \text{ for power function)}$$

where, again, Y_{ci} = calculated value of dependent variable at point i,

Y_i = observed value of dependent variable at point i.

If there is a solution in the region defined above, then that solution represents an absolute minimum of Q and is the only solution in that region. In addition, it seems reasonable to assume that the above condition should hold in order to have a good fit for the power and exponential functions.

In summary, one should be aware that for a nonlinear function the "solution" obtained may not represent an absolute minimum for Q. The only sure way to know is to try all combinations of the parameters for each data sample to determine all "solutions" and to then determine

---

*H. O. Hartley, "The Modified Gauss-Newton Method for the Fitting of Non-linear Regression Functions by Least-squares," Technometrics, Vol. 3, No. 2, May 1961, pp. 273-274.

**The proof is given in Appendix D of RM-4879-PR by Graver, Boren, op. cit.

which solution gives the lowest sum of squares of Y residuals. For practical reasons this is very difficult to do. However, one must remember that he is attempting to find a solution to a function that adequately represents the data. Whether or not there are solutions in other unknown regions may be rather unimportant if the solution that is found is satisfactory to the analyst--that is, it satisfies his criterion for a good fit.*

---

*For further information on nonlinear least-squares solutions the reader is referred to Applied Regression Analysis, by N. R. Draper and H. Smith, John Wiley and Sons, Inc., New York, London, Sydney, 1966, Chap. 10, pp. 263-304.

Appendix B

LEAST-SQUARES SOLUTION FOR ASYMPTOTIC-POWER FUNCTION

DETERMINATION OF PARAMETERS A, B, C

For this program, a least-squares fit is assumed for the data, in which the sum of the squares of the Y residuals (differences between the observed values of Y and the corresponding calculated values of Y from the regression equation) is a minimum. Therefore, the Y residual (R) at any point i is

$$R_i = Y_i - Y_{ci},$$

or

$$R_i = Y_i - (A \cdot X1_i^B + C), \qquad (1)$$

where

R_i = Y residual at point i,

Y_i = observed value of Y at point i,

Y_{ci} = calculated value of Y at point i,

$X1_i$ = value of independent variable at point i,

A, B, C = parameters to be determined.

The requirement for a least-squares fit for N sets of points is that the sum of the squares of the Y residuals (denoted by Q) shall be a minimum; that is,

$$Q = \sum_{i=1}^{N} (Y_i - A \cdot X1_i^B - C)^2 = \text{min}. \qquad (2)$$

First let $Z_i = X1_i^B$. Then Equation (2) can be written as follows:

$$Q = \sum_{i=1}^{N} (Y_i - A \cdot Z_i - C)^2 = \min., \tag{3}$$

From normal regression equations*

$$A = \frac{\sum_{i=1}^{N} (Y_i - \overline{Y}) \cdot (Z_i - \overline{Z})}{\sum_{i=1}^{N} (Z_i - \overline{Z})^2}, \tag{4}$$

$$C = \overline{Y} - A \cdot \overline{Z}. \tag{5}$$

Substituting the above expression for C into Equation (3) gives:

$$Q = \sum_{i=1}^{N} [Y_i - A \cdot Z_i - (\overline{Y} - A \cdot \overline{Z})]^2 ,$$

$$Q = \sum_{i=1}^{N} [(Y_i - \overline{Y}) - A \cdot (Z_i - \overline{Z})]^2 ,$$

$$Q = \sum_{i=1}^{N} (Y_i - \overline{Y})^2 - 2 \cdot A \cdot \sum_{i=1}^{N} [(Y_i - \overline{Y})$$

$$\cdot (Z_i - \overline{Z})] + A^2 \cdot \sum_{i=1}^{N} (Z_i - \overline{Z})^2 .$$

Let

$$S_{yy} = \sum_{i=1}^{N} (Y_i - \overline{Y})^2 ,$$

---

*Overscores are used to denote means.

$$S_{yz} = \sum_{i=1}^{N} (Y_i - \bar{Y}) \cdot (Z_i - \bar{Z}),$$

$$S_{zz} = \sum_{i=1}^{N} (Z_i - \bar{Z})^2 .$$

Therefore,

$$Q = S_{yy} - 2 \cdot A \cdot S_{yz} + A^2 \cdot S_{zz} .$$

From Equation (4) and the above definitions:

$$A = \frac{S_{yz}}{S_{zz}} , \tag{6}$$

Thus,

$$Q = S_{yy} - 2 \cdot \left(\frac{S_{yz}}{S_{zz}}\right) \cdot S_{yz} + \left(\frac{S_{yz}}{S_{zz}}\right)^2 \cdot S_{zz} ,$$

$$Q = S_{yy} - \frac{(S_{yz})^2}{S_{zz}} . \tag{7}$$

However,

$$S_{yz} = \sum_{i=1}^{N} (Y_i - \bar{Y}) \cdot (Z_i - \bar{Z}) ,$$

$$S_{yz} = \sum_{i=1}^{N} (Y_i - \bar{Y}) \cdot Z_i - \bar{Z} \cdot \sum_{i=1}^{N} (Y_i - \bar{Y}) ,$$

$$S_{yz} = \sum_{i=1}^{N} (Y_i - \overline{Y}) \cdot Z_i, \qquad \text{since} \quad \sum_{i=1}^{N} (Y_i - \overline{Y}) = 0.$$

Also,

$$S_{zz} = \sum_{i=1}^{N} (Z_i - \overline{Z})^2,$$

$$S_{zz} = \sum_{i=1}^{N} Z_i^2 - 2 \cdot \overline{Z} \cdot \sum_{i=1}^{N} Z_i + \sum_{i=1}^{N} \overline{Z}^2,$$

$$S_{zz} = \sum_{i=1}^{N} Z_i^2 - 2 \cdot \left(\frac{1}{N} \cdot \sum_{i=1}^{N} Z_i\right) \cdot \sum_{i=1}^{N} Z_i$$

$$+ N \cdot \left(\frac{1}{N} \cdot \sum_{i=1}^{N} Z_i\right)^2 ,$$

$$S_{zz} = \sum_{i=1}^{N} Z_i^2 - \frac{1}{N} \cdot \left(\sum_{i=1}^{N} Z_i\right)^2 .$$

Finally, using $X1_i^B$ for Z_i, and the above definitions, one has, from Equations (6), (5), and (7):

$$A = \frac{\sum_{i=1}^{N} (Y_i - \overline{Y}) \cdot X1_i^B}{\sum_{i=1}^{N} X1_i^{2B} - \frac{1}{N} \cdot \left(\sum_{i=1}^{N} X1_i^B\right)^2} , \tag{8}$$

$$C = \overline{Y} - \frac{1}{N} \cdot A \cdot \sum_{i=1}^{N} X1_i^B , \tag{9}$$

$$Q = \sum_{i=1}^{N} (Y_i - \bar{Y})^2 - \frac{\left[\sum_{i=1}^{N} (Y_i - \bar{Y}) \cdot X1_i^B\right]^2}{\sum_{i=1}^{N} X1_i^{2B} - \frac{1}{N} \cdot \left(\sum X1_i^B\right)^2} = \min. \tag{10}$$

Equation (10), which represents the sum of squares of Y residuals, is therefore equivalent to Equation (2) but is expressed in terms of only the one parameter B.

If Q is to be a minimum, the partial derivatives of Q with respect to the parameters must be zero.* However, since A and C are determined from Equations (8) and (9), there is no need to obtain the partial derivatives of Q with respect to A and C. From Equation (2), one has:

$$\frac{\partial Q}{\partial B} = 2 \cdot \sum_{i=1}^{N} (Y_i - A \cdot X1_i^B - C) \cdot (-A \cdot X1_i^B \cdot \ln X1_i) = 0 ,$$

$$\sum_{i=1}^{N} Y_i \cdot X1_i^B \cdot \ln X1_i - A \cdot \sum_{i=1}^{N} X1_i^{2B} \cdot \ln X1_i - C \cdot \sum_{i=1}^{N} X1_i^B \cdot \ln X1_i = 0.$$

Let G represent the above function. Therefore,

$$G = \sum_{i=1}^{N} Y_i \cdot X1_i^B \cdot \ln X1_i - A \cdot \sum_{i=1}^{N} X1_i^{2B} \cdot \ln X1_i - C \sum_{i=1}^{N} X1_i^B \cdot \ln X1_i . \tag{11}$$

The problem then becomes one of finding the value of B that makes G zero.

---

*As was stated previously, this is not a sufficient condition for a function that is not linear with respect to all of its parameters. (See Appendix A.)

The sequence of operations in the computer program is as follows.*
First, the various summations involved in Equations (8), (9), and (11) are obtained using B = -3.96 (initially). Then A and C are determined from Equations (8) and (9), respectively.** After these calculations are made, the value of G is obtained, and its algebraic sign is noted. The machine then steps the value of B by +0.05, repeats all of the summations and calculations, and checks the algebraic sign of G again. This procedure is continued until the algebraic sign of G is reversed, signifying that a solution lies somewhere between the previous value of B and the value of B at this cross-over point.

At this point, the program begins an iterative operation in which at each cross-over point the incremental step is halved and the direction of advance is reversed. This iterative procedure is done as many times as desired to give any degree of accuracy required for B. In the program, this procedure is repeated until the changes in the absolute values of A, B, and C from one iteration to the next are each equal to, or less than, 10^{-8}.

The search for roots continues to B = -0.01. After this point is reached, the program begins another search starting at B = +0.01 and proceeding by increments of +0.05 out to +3.96. If no solution at all is found within these limits, a statement to this effect is printed, and the program continues on to the next run. Any time a solution is found for A, B, and C, the sum of squares of Y residuals (Q) is determined and compared with the corresponding value for the previous solution (if there was one). The solution that gives the lowest sum of squares of Y residuals is stored temporarily for comparison with any future solution so obtained. In this way, when the search is completed and if there is a solution, that solution will generally represent the lowest sum of squares of Y residuals in the region searched.

---

*Acknowledgment is made to Mr. James Johnston (now at the Institute for Defense Analyses) for his suggestions in the initial programming aspects of this problem.

**If C is specified, then that value is used instead of calculating C from Equation (9).

Any "solution" found in the specified range for B represents a solution for which the partial derivatives of Q with respect to the parameters are zero. The Q value for that solution is also compared with the Q values for the end points of B to make sure that Q is not decreasing to some other minimum outside the range of B. Test calculations of Q as represented by Equation (10) indicate that Q increases smoothly to apparently constant values for large values of negative or positive B. As of now, the author has not been able to determine any requirements for unimodality of Q but has observed that for data applicable to this function, Q seems to be unimodal in the region searched. Even if it is not, the most minimum of those modes will usually be found. As stated before, one must be aware that there could be other minima outside the range searched which cannot be determined by the above method. However, this may be relatively unimportant if the "solution" found satisfies the analyst's criterion for a good fit.

The above limits on B and the incremental stepping value of 0.05 were chosen on the basis of economic computer operating time and the extent to which the search range should be covered in order to lessen the chances of missing a root. Although two roots could conceivably be missed in the incremental step of 0.05, indicating that the G function goes from, say, a positive to a negative to a positive value within an interval of B equal to 0.05, this seems rather unlikely. Such a function would have to behave extremely erratically, and test results seem to indicate that this function does not generally behave in this manner.

Perhaps it should be noted that a degenerate, or trivial, case results if B = 0 or if all Y values are constant or all X1 values are constant. Any of these conditions causes the numerator and denominator of Equation (8) to be zero (an indeterminate condition). Of course, this can be seen from the asymptotic-power equation itself. Any of the above conditions cause it to reduce to:

$$Y = \text{constant}.$$

Appendix C

LISTING OF CURVES FORTRAN-IV COMPUTER PROGRAM

```
$IBFTC MAIN
      COMMON         A, A1, B, B1, C, C1, D, D1, AH, BH, CH, DH, AN, AN1,
     C               AN2, DF1, DF2, DFT, FVALUE, CO, CV, R, PDEVM, SEY,
     C               XV, YV, YDEVSQ, EA
      COMMON         I4, IDISK, IERR, IFV, ISOLVE, ISC1, N, NIV, NOTE, NP
      COMMON         H(4,4), T(4), SDEV(4), VMEAN(4), S(30), V(201,6),
     C               PDEV(200), X1L(200), X2L(200), X3L(200), YL(200),
     C               YC(200), YDEV(200)
      COMMON         ISC(4), JX(6), FMT(18), TITLE(16), IND, IPAGE, IEQ,
     C               IORD, NID, NRUN
      DIMENSION      ID1(201), ID2(201), Y(201), X1(201), X2(201),
     C               X3(201), F(1)
      EQUIVALENCE    (F(1), A), (ID1(1), V(1,1)), (ID2(1), V(1,2)),
     C               (Y(1), V(1,3)), (X1(1), V(1,4)), (X2(1), V(1,5)),
     C               (X3(1), V(1,6)), (JX(1), J1), (JX(2), J2),
     C               (JX(3), J3), (JX(4), J4), (JX(5), J5), (JX(6), J6)
C
C     MAIN ROUTINE OF 'CURVES' REGRESSION ANALYSIS COMPUTER PROGRAM
C
C     SET PAGE NUMBER TO 1.
   10 IPAGE = 1
C     SET RUN COUNTER TO 1.
      NRUN = 1
C     PRINT MAIN HEADING ON FIRST PAGE.
      WRITE (6, 20) IPAGE
   20 FORMAT (1H1/, 45X, 43HCURVES REGRESSION ANALYSIS COMPUTER PROGRAM,
     C 27X, 5HPAGE , I2 //// )
C     STEP PAGE NUMBER BY 1.
      IPAGE = IPAGE + 1
C     CLEAR COMMON.
   30 DO 40 I = 1, 2706
      F(I) = 0.
   40 CONTINUE
C     SET SUBROUTINE INDICATOR TO 1.
      IND = 1
      CALL READ
      CAL  INPUT
      CALL CHECK
C     CHECK ERROR DESIGNATOR.
      IF (IERR .EQ. 1) GO TO 110
      CALL SUMS
      GO TO (50, 60, 70, 80, 90), IEQ
   50 CALL LINE
      GO TO 100
   60 CALL PARA
      GO TO 100
   70 CALL POWR
      GO TO 100
   80 CALL ASYM
      GO TO 100
   90 CALL EXPO
  100 IF (IERR .EQ. 1) GO TO 110
      CALL STAT
      CALL OUT1
      CALL OUT2
C     STEP RUN COUNTER BY 1.
```

```
  110 NRUN = NRUN + 1
C     RESET PAGE NUMBER TO 1
      IPAGE = 1
      GO TO 30
      END
```

```
$IBFTC READ
      SUBROUTINE READ
      COMMON        A, A1, B, B1, C, C1, D, D1, AH, BH, CH, DH, AN, AN1,
     C              AN2, DF1, DF2, DFT, FVALUE, CO, CV, R, PDEVM, S ,
     C              XV, YV, YDEVSQ, EA
      COMMON        IA, IDISK, IERR, IFV, ISOLVE, ISC1, N, NIV, NOTE, NP
      COMMON        H(4,4), T(4), SDEV(4), VMFAN(4), S(30), V(201,6),
     C              PDEV(200), X1L(200), X2L(200), X3L(200), YL(200),
     C              YC(200), YDEV(200)
      COMMON        ISC(4), JX(6), FMT(18), TITLE(16), IND, IPAGE, IEQ,
     C              IORD, NID, NRUN
      DIMENSION     ID1(201), ID2(201), Y(201), X1(201), X2(201),
     C              X3(201), F(1)
      DIMENSION     KX(5), KV(6)
      EQUIVALENCE   (F(1), A), (ID1(1), V(1,1)), (ID2(1), V(1,2)),
     C              (Y(1), V(1,3)), (X1(1), V(1,4)), (X2(1), V(1,5)),
     C              (X3(1), V(1,6)), (JX(1), J1), (JX(2), J2),
     C              (JX(3), J3), (JX(4), J4), (JX(5), J5), (JX(6), J6)
      DATA          KV(1), KV(2), KV(3), KV(4), KV(5), KV(6) /
     C              2HID, 2HY1, 2HX1, 2HX2, 2HX3, 2H  /
C
C     SUBROUTINE FOR READING TITLE CARD, ORDER CARD, AND VARIABLE-FORMAT
C      CARD
C
C     READ TITLE CARD.
   10 READ (5, 20) IFMT, IDISK, IEQ1, IORD1, ISC1, IA,
     C (TITLE(I), I = 1, 16)
   20 FORMAT (6I1, 2X, 16A4)
C     CHECK TITLE CARD FOR TERMINATION DESIGNATOR (IFMT = 9)
   30 IF (IFMT .EQ. 9) CALL EXIT
      IF (IEQ1 .GE. 1) IEQ = IEQ1
      IF (IORD1 .GT. 0) IORD = 1 - (IORD1/2)
C     PRINT TITLE AND PAGE NUMBER ON FIRST PAGE
   40 IF (NRUN .GT. 1) GO TO 55
      WRITE (6, 50) (TITLE(I), I = 1, 16)
   50 FORMAT (1H / 10X, 16A4 /// )
      GO TO 65
   55 WRITE (6, 60) (TITLE(I), I = 1, 16), IPAGE
   60 FORMAT (1H1/ 10X, 16A4, 41X, 5HPAGE , I2 /// )
C     STEP PAGE NUMBER BY 1.
      IPAGE = IPAGE + 1
C     PRINT SUBHEADING.
   65 CALL PRINT
C     TEST WHETHER TITLE CARD FOR FIRST RUN CONTAINS THE VARIABLE-FORMAT
C      INDICATOR IN COLUMN 1.
      IF (NRUN .GT. 1) GO TO 90
      IF (IFMT .GT. 0) GO TO 100
C     WRITE ERROR MESSAGE.
   70 WRITE (6, 80)
   80 FORMAT (1HO//, 10X, 118HTHE VARIABLE-FORMAT INDICATOR HAS NOT BEEN
     C ENTERED IN COLUMN 1 OF THE FIRST TITLE CARD.  THIS JOB HAS BEEN T
     CERMINATED.)
      CALL EXIT
C     TEST WHETHER ORDER CARD AND FORMAT CARD ARE TO BE READ FOR THIS
C      RUN.
   90 IF (IFMT .EQ. 0) GO TO 170
```

```
C     READ ORDER CARD.
  100 READ (5, 110) (KX(I), I = 1, 5), NID1
  110 FORMAT (5(A2, 1X), I1)
      IF (NID1 .GT. 0) NID = NID1
      I = 1
      JMAX = 1
      DO 150 K = 1, 5
      DO 120 J = 1, 6
      IF (KX(K) .EQ. KV(J)) GO TO 130
  120 CONTINUE
C     WRITE ERROR MESSAGE.
      WRITE (6, 125)
  125 FORMAT (1HO// 10X,      98HTHERE IS AN ERROR IN THE VARIABLE-FORMAT
     C INDEX (SECOND INPUT CARD).  THIS JOB HAS BEEN TERMINATED. )
      CALL EXIT
  130 IF (J .NE. 1) GO TO 140
      JX(I) = 1
      JX(I+1) = 2
      I = I + 2
      GO TO 145
  140 JX(I) = J + 1
      I = I + 1
  145 IF (J .EQ. 6) GO TO 150
      IF (JMAX .LT. J) JMAX = J
  150 CONTINUE
C     READ FORMAT CARD
      READ (5, 160) (FMT(I), I = 1, 18)
  160 FORMAT (18A4)
C     NUMBER OF INDEPENDENT VARIABLES
  170 NIV = JMAX - 2
C     NUMBER OF PARAMETERS TO BE SOLVED (NP)
      NP = NIV + 1 - IA
      IF (IEQ .EQ. 2 .OR. IEQ .EQ. 4) NP = NP + 1
      IF (IA .EQ. 0 .AND. ISC1 .EQ. 0) RETURN
C     READ SCALE FACTORS AND/OR VALUE OF Y INTERCEPT.
      IF (IEQ .NE. 4) READ (5, 180) (ISC(I), I = 1, 4), A
      IF (IEQ .EQ. 4) READ (5, 180) (ISC(I), I = 1, 4), C
  180 FORMAT (4I2, 8X, F10.0)
      RETURN
      END
```

```
$IBFTC PRINT
       SUBROUTINE PRINT
       COMMON         A, A1, B, B1, C, C1, D, D1, AH, BH, CH, DH, AN, AN1,
      C               AN2, DF1, DF2, DFT, FVALUE, CD, CV, R, PDEVM, SEY,
      C               XV, YV, YDEVSQ, EA
       COMMON         IA, IDISK, IERR, IFV, ISOLVE, ISC1, N, NIV, NOTE, NP
       COMMON         H(4,4), T(4), SDEV(4), VMEAN(4), S(30), V(201,6),
      C               PDEV(200), X1L(200), X2L(200), X3L(200), YL(200),
      C               YC(200), YDEV(200)
       COMMON         ISC(4), JX(6), FMT(18), TITLE(16), IND, IPAGE, IEQ,
      C               IORD, NID, NRUN
       DIMENSION      ID1(201), ID2(201), Y(201), X1(201), X2(201),
      C               X3(201), F(1)
       EQUIVALENCE    (F(1), A), (ID1(1), V(1,1)), (ID2(1), V(1,2)),
      C               (Y(1), V(1,3)), (X1(1), V(1,4)), (X2(1), V(1,5)),
      C               (X3(1), V(1,6)), (JX(1), J1), (JX(2), J2),
      C               (JX(3), J3), (JX(4), J4), (JX(5), J5), (JX(6), J6)
C
C      SUBROUTINE FOR PRINTING SUBHEADINGS
C
   10 GO TO (20, 40, 60, 80, 100), IEQ
   20 WRITE (6, 30)
   30 FORMAT (1H0/, 36X, 59HLINEAR REGRESSION -- Y = A + (B * X1) + (C *
     C X2) + (D * X3) / )
      RETURN
   40 WRITE (6, 50)
   50 FORMAT (1H0/, 39X,    52HPARABOLIC REGRESSION -- Y = A + (B * X) +
     C (C * X**2) / )
      RETURN
   60 WRITE (6, 70)
   70 FORMAT (1H0/ 37X, 55HPOWER REGRESSION -- Y = A * (X1**B) * (X2**C)
     C * (X3**D) / )
      RETURN
   80 WRITE (6, 90)
   90 FORMAT (1H0/, 39X,      51HASYMPTOTIC-POWER REGRESSION -- Y = (A *
     C (X**B)) + C / )
      RETURN
  100 WRITE (6, 110)
  110 FORMAT (1H0/, 42X,       46HEXPONENTIAL REGRESSION -- Y = EXP(A +
     + (B * X)) / )
      RETURN
      END
```

```
$IBFTC INPUT
      SUBROUTINE INPUT
      COMMON         A, A1, B, B1, C, C1, D, D1, AH, BH, CH, DH, AN, AN1,
     C               AN2, DF1, DF2, DFT, FVALUE, CD, CV, R, PDEVM, SEY,
     C               XV, YV, YDEVSQ, EA
      COMMON         IA, IDISK, IERR, IFV, ISOLVE, ISC1, N, NIV, NOTE, NP
      COMMON         H(4,4), T(4), SDEV(4), VMEAN(4), S(30), V(201,6),
     C               PDEV(200), X1L(200), X2L(200), X3L(200), YL(200),
     C               YC(200), YDEV(200)
      COMMON         ISC(4), JX(6), FMT(18), TITLE(16), IND, IPAGE, IEQ,
     C               IORD, NID, NRUN
      DIMENSION      ID1(201), ID2(201), Y(201), X1(201), X2(201),
     C               X3(201), F(1)
      DIMENSION      VDATA(20)
      EQUIVALENCE    (F(1), A), (ID1(1), V(1,1)), (ID2(1), V(1,2)),
     C               (Y(1), V(1,3)), (X1(1), V(1,4)), (X2(1), V(1,5)),
     C               (X3(1), V(1,6)), (JX(1), J1), (JX(2), J2),
     C               (JX(3), J3), (JX(4), J4), (JX(5), J5), (JX(6), J6)
      DATA           BLANK, IBLANK/4H    , 4H    /
C
C     SUBROUTINE FOR READING IN DATA AND WRITING DATA ONTO UTILITY DISK
C
   10 IF (IDISK .EQ. 0) GO TO 60
      REWIND 4
C     SET INPUT TAPE NUMBER
      IM = 4
      IF (IDISK .EQ. 2) GO TO 70
      DO 50 I = 1, 201
C     READ INPUT DATA AS ALPHANUMERIC DATA
      READ (5, 20) (VDATA(J), J = 1, 20)
   20 FORMAT (20A4)
C     WRITE INPUT DATA ONTO UTILITY DISK.
      WRITE (4, 20) (VDATA(J), J = 1, 20)
C     CHECK FOR BLANK CARD.
      DO 30 K = 1, 20
      IF (VDATA(K) .NE. BLANK) GO TO 50
   30 CONTINUE
      REWIND 4
   40 GO TO 70
   50 CONTINUE
      REWIND 4
      GO TO 70
C     SET INPUT TAPE NUMBER.
   60 IM = 5
C     READ INPUT DATA FROM EITHER CARDS (IDISK = 0) OR FROM UTILITY
C      DISK (IDISK = 1).
   70 IREAD = 0
      DO 80 I = 1, 6
      IF (JX(I) .EQ. 1) IREAD = 1
   80 CONTINUE
C     SET IDENTIFIERS TO BLANK
      DO 85 I = 1, 200
      ID1(I) = IBLANK
      ID2(I) = IBLANK
   85 CONTINUE
      IF (IREAD .EQ. 1) GO TO 90
```

```
      GO TO (100, 130, 160), NIV
   90 IF (MID .EQ. 1) GO TO 280
      GO TO (190, 220, 250), NIV
  100 DO 120 I = 1, 201
  110 READ (IM, FMT) V(I,J1), V(I,J2)
      IF (ABS(Y(I)) + ABS(X1(I)) .EQ. 0.) GO TO 580
      IF (Y(I) .EQ. 0. .OR. X1(I) .EQ. 0.) GO TO 110
  120 CONTINUE
      GO TO 590
  130 DO 150 I = 1, 201
  140 READ (IM, FMT) V(I,J1), V(I,J2), V(I,J3)
      IF (ABS(Y(I)) + ABS(X1(I)) + ABS(X2(I)) .EQ. 0.) GO TO 580
      IF (Y(I) .EQ. 0. .OR. X1(I) .EQ. 0. .OR. X2(I) .EQ. 0.) GO TO 140
  150 CONTINUE
      GO TO 590
  160 DO 180 I = 1, 201
  170 READ (IM, FMT) V(I,J1), V(I,J2), V(I,J3), V(I,J4)
      IF (ABS(Y(I)) + ABS(X1(I)) + ABS(X2(I)) + ABS(X3(I)) .EQ. 0.)
     C GO TO 580
      IF (Y(I) .EQ. 0. .OR. X1(I) .EQ. 0. .OR. X2(I) .EQ. 0. .OR.
     C X3(I) .EQ. 0.) GO TO 170
  180 CONTINUE
      GO TO 590
  190 DO 210 I = 1, 201
  200 READ (IM, FMT) V(I,J1), V(I,J2), V(I,J3), V(I,J4)
      IF (ABS(Y(I)) + ABS(X1(I)) .EQ. 0.) GO TO 580
      IF (Y(I) .EQ. 0.  .OR. X1(I) .EQ. 0.) GO TO 200
  210 CONTINUE
      GO TO 590
  220 DO 240 I = 1, 201
  230 READ (IM, FMT) V(I,J1), V(I,J2), V(I,J3), V(I,J4)
      IF (ABS(Y(I)) + ABS(X1(I)) + ABS(X2(I)) .EQ. 0.) GO TO 580
      IF (Y(I) .EQ. 0. .OR. X1(I) .EQ. 0. .OR. X2(I) .EQ. 0.) GO TO 230
  240 CONTINUE
      GO TO 590
  250 DO 270 I = 1, 201
  260 READ (IM, FMT) V(I,J1), V(I,J2), V(I,J3), V(I,J4), V(I,J5),V(I,J6)
      IF (ABS(Y(I)) + ABS(X1(I)) + ABS(X2(I)) + ABS(X3(I)) .EQ. 0.)
     C GO TO 580
      IF (Y(I) .EQ. 0. .OR. X1(I) .EQ. 0. .OR. X2(I) .EQ. 0. .OR.
     C X3(I) .EQ. 0.) GO TO 260
  270 CONTINUE
  280 DO 290 K = 1, 5
      IF (JX(K) .EQ. 1) GO TO 300
  290 CONTINUE
      GO TO 610
  300 GO TO (310, 380, 470), NIV
  310 GO TO (320, 340, 360), K
  320 DO 330 I = 1, 201
  325 READ (IM, FMT) V(I,J1), V(I,J3), V(I,J4)
      IF (ABS(Y(I)) + ABS(X1(I)) .EQ. 0.) GO TO 580
      IF (Y(I) .EQ. 0. .OR. X1(I) .EQ. 0.) GO TO 325
  330 CONTINUE
      GO TO 590
  340 DO 350 I = 1, 201
  345 READ (IM, FMT) V(I,J1), V(I,J2), V(I,J4)
```

```
      IF (ABS(Y(I)) + ABS(X1(I)) .EQ. 0.) GO TO 580
      IF (Y(I) .EQ. 0. .OR. X1(I) .EQ. 0.) GO TO 345
  350 CONTINUE
      GO TO 590
  360 DO 370 I = 1, 201
  365 READ (IM, FMT) V(I,J1), V(I,J2), V(I,J3)
      IF (ABS(Y(I)) + ABS(X1(I)) .EQ. 0.) GO TO 580
      IF (Y(I) .EQ. 0. .OR. X1(I) .EQ. 0.) GO TO 365
  370 CONTINUE
      GO TO 590
  380 GO TO (390, 410, 430, 450), K
  390 DO 400 I = 1, 201
  395 READ (IM, FMT) V(I,J1), V(I,J3), V(I,J4), V(I,J5)
      IF (ABS(Y(I)) + ABS(X1(I)) + ABS(X2(I)) .EQ. 0.) GO TO 580
      IF (Y(I) .EQ. 0. .OR. X1(I) .EQ. 0. .OR. X2(I) .EQ. 0.) GO TO 395
  400 CONTINUE
      GO TO 590
  410 DO 420 I = 1, 201
  415 READ (IM, FMT) V(I,J1), V(I,J2), V(I,J4), V(I,J5)
      IF (ABS(Y(I)) + ABS(X1(I)) + ABS(X2(I)) .EQ. 0.) GO TO 580
      IF (Y(I) .EQ. 0. .OR. X1(I) .EQ. 0. .OR. X2(I) .EQ. 0.) GO TO 415
  420 CONTINUE
      GO TO 590
  430 DO 440 I = 1, 201
  435 READ (IM, FMT) V(I,J1), V(I,J2), V(I,J3), V(I,J5)
      IF (ABS(Y(I)) + ABS(X1(I)) + ABS(X2(I)) .EQ. 0.) GO TO 580
      IF (Y(I) .EQ. 0. .OR. X1(I) .EQ. 0. .OR. X2(I) .EQ. 0.) GO TO 435
  440 CONTINUE
      GO TO 590
  450 DO 460 I = 1, 201
  455 READ (IM, FMT) V(I,J1), V(I,J2), V(I,J3), V(I,J4)
      IF (ABS(Y(I)) + ABS(X1(I)) + ABS(X2(I)) .EQ. 0.) GO TO 580
      IF (Y(I) .EQ. 0. .OR. X1(I) .EQ. 0. .OR. X2(I) .EQ. 0.) GO TO 455
  460 CONTINUE
      GO TO 590
  470 GO TO (480, 500, 520, 540, 560), K
  480 DO 490 I = 1, 201
  485 READ (IM, FMT) V(I,J1), V(I,J3), V(I,J4), V(I,J5), V(I,J6)
      IF (ABS(Y(I)) + ABS(X1(I)) + ABS(X2(I)) + ABS(X3(I)) .EQ. 0.)
     C GO TO 580
      IF (Y(I) .EQ. 0. .OR. X1(I) .EQ. 0. .OR. X2(I) .EQ. 0. .OR.
     C X3(I) .EQ. 0.) GO TO 485
  490 CONTINUE
      GO TO 590
  500 DO 510 I = 1, 201
  505 READ (IM, FMT) V(I,J1), V(I,J2), V(I,J4), V(I,J5), V(I,J6)
      IF (ABS(Y(I)) + ABS(X1(I)) + ABS(X2(I)) + ABS(X3(I)) .EQ. 0.)
     C GO TO 580
      IF (Y(I) .EQ. 0. .OR. X1(I) .EQ. 0. .OR. X2(I) .EQ. 0. .OR.
     C X3(I) .EQ. 0.) GO TO 505
  510 CONTINUE
      GO TO 590
  520 DO 530 I = 1, 201
  525 READ (IM, FMT) V(I,J1), V(I,J2), V(I,J3), V(I,J5), V(I,J6)
      IF (ABS(Y(I)) + ABS(X1(I)) + ABS(X2(I)) + ABS(X3(I)) .EQ. 0.)
     C GO TO 580
```

```
      IF (Y(I) .EQ. 0. .OR. X1(I) .EQ. 0. .OR. X2(I) .EQ. 0. .OR.
     C X3(I) .EQ. 0.) GO TO 525
  530 CONTINUE
      GO TO 590
  540 DO 550 I = 1, 201
  545 READ (IM, FMT) V(I,J1), V(I,J2), V(I,J3), V(I,J4), V(I,J6)
      IF (ABS(Y(I)) + ABS(X1(I)) + ABS(X2(I)) + ABS(X3(I)) .EQ. 0.)
     C GO TO 580
      IF (Y(I) .EQ. 0. .OR. X1(I) .EQ. 0. .OR. X2(I) .EQ. 0. .OR.
     C X3(I) .EQ. 0.) GO TO 545
  550 CONTINUE
      GO TO 590
  560 DO 570 I = 1, 201
  565 READ (IM, FMT) V(I,J1), V(I,J2), V(I,J3), V(I,J4), V(I,J5)
      IF (ABS(Y(I)) + ABS(X1(I)) + ABS(X2(I)) + ABS(X3(I)) .EQ. 0.)
     C GO TO 580
      IF (Y(I) .EQ. 0. .OR. X1(I) .EQ. 0. .OR. X2(I) .EQ. 0. .OR.
     C X3(I) .EQ. 0.) GO TO 565
  570 CONTINUE
      GO TO 590
  580 IF (IDISX .NE. 0) REWIND 4
C     SET N EQUAL TO NUMBER OF DATA POINTS.
      N = I - 1
C     FLOAT N.
      AN = N
C     NUMBER OF DATA POINTS LESS 1
      AN1 = AN - 1.0
C     NUMBER OF DATA POINTS LESS 2
      AN2 = AN - 2.0
      RETURN
  590 WRITE (6, 600)
  600 FORMAT (1H0// 10X,      96HNUMBER OF INPUT DATA POINTS HAS EXCEEDED
     C MAXIMUM ALLOWABLE (200).  THIS JOB HAS BEEN TERMINATED.)
      CALL EXIT
  610 WRITE (6, 620)
  620 FORMAT (1H0// 10X,        111HTHERE IS AN ERROR ASSOCIATED WITH THE
     C IDENTIFIER-FORMAT DESIGNATOR (ORDER CARD).  THIS JOB HAS BEEN TER
     CMINATED.)
      CALL EXIT
      END
```

```
$IBFTC CHECK
      SUBROUTINE CHECK
      COMMON        A, A1, B, B1, C, C1, D, D1, AH, BH, CH, DH, AN, AN1,
     C              AN2, DF1, DF2, DFT, FVALUE, CO, CV, R, PDFVM, SEY,
     C              XV, YV, YDEVSQ, EA
      COMMON        IA, IDISK, IERR, IEV, ISOLVE, ISCL, N, NIV, NOTE, NP
      COMMON        H(4,4), T(4), SDEV(4), VMEAN(4), S(30), V(201,6),
     C              PDEV(200), X1L(200), X2L(200), X3L(200), YL(200),
     C              YC(200), YDEV(200)
      COMMON        ISC(4), JX(6), FMT(18), TITLE(16), IND, IPAGE, IEO,
     C              ICRD, NID, NRUN
      DIMENSION     ID1(201), ID2(201), Y(201), X1(201), X2(201),
     C              X3(201), F(1)
      DIMENSION     RSC(4)
      EQUIVALENCE   (F(1), A), (ID1(1), V(1,1)), (ID2(1), V(1,2)),
     C              (Y(1), V(1,3)), (X1(1), V(1,4)), (X2(1), V(1,5)),
     C              (X3(1), V(1,6)), (JX(1), J1), (JX(2), J2),
     C              (JX(3), J3), (JX(4), J4), (JX(5), J5), (JX(6), J6)
C
C     SUBROUTINE FOR CHECKING DATA FOR ERRORS AND SCALING DATA
C
C     CALCULATE SCALE FACTORS.
   10 DO 20 I = 1, 4
      RSC(I) = 1.0
      IF (ISC(I) .NE. 0) RSC(I) = 10.0**(ISC(I))
   20 CONTINUE
   30 GO TO (40, 60, 80), NIV
   40 DO 50 I = 1, N
      IF (Y(I) .LE. 0. .OR. X1(I) .LE. 0.) GO TO 170
      DO 45 J = 3, 6
      V(I,J) = V(I,J) * RSC(J-2)
   45 CONTINUE
   50 CONTINUE
      GO TO 120
   60 DO 70 I = 1, N
      IF (Y(I) .LE. 0. .OR. X1(I) .LE. 0. .OR. X2(I) .LE. 0.) GO TO 170
      DO 65 J = 3, 6
      V(I,J) = V(I,J) * RSC(J-2)
   65 CONTINUE
   70 CONTINUE
      GO TO 120
   80 DO 90 I = 1, N
      IF (Y(I) .LE. 0. .OR. X1(I) .LE. 0. .OR. X2(I) .LE. 0. .OR.
     C X3(I) .LE. 0.) GO TO 170
      DO 85 J = 3, 6
      V(I,J) = V(I,J) * RSC(J-2)
   85 CONTINUE
   90 CONTINUE
  120 IF (ISC(1) .NE. 0) WRITE (6, 130) RSC(1)
      IF (ISC(2) .NE. 0) WRITE (6, 140) RSC(2)
      IF (ISC(3) .NE. 0) WRITE (6, 150) RSC(3)
      IF (ISC(4) .NE. 0) WRITE (6, 160) RSC(4)
  130 FORMAT (1H , 9X, 20HY SCALE FACTOR      , E8.1 )
  140 FORMAT (1H , 9X, 20HX1 SCALE FACTOR     , E8.1 )
  150 FORMAT (1H , 9X, 20HX2 SCALE FACTOR     , E8.1 )
  160 FORMAT (1H , 9X, 20HX3 SCALE FACTOR     , E8.1 )
```

```
      RETURN
C     ERROR MESSAGE
  170 WRITE (6, 180) I
  180 FORMAT (1H0// 10X, 81HA ZERO OR NEGATIVE VALUE EXISTS IN THE INPUT
     C DATA (FOR EXAMPLE, DATA CARD NUMBER , I3,33H).  THIS RUN HAS BEEN
     C TERMINATED. )
      IERR = 1
      RETURN
      END
```

```
$IBFTC SUMS
      SUBROUTINE SUMS
      COMMON          A, A1, B, B1, C, C1, D, D1, AH, BH, CH, DH, AN, AN1,
     C                AN2, DF1, DF2, DFT, FVALUE, CD, CV, R, PDEVM, SEY,
     C                XV, YV, YDEVSQ, EA
      COMMON          IA, IDISK, IERR, IFV, ISOLVE, ISC1, M, NIV, NOTE, NP
      COMMON          H(4,4), T(4), SDEV(4), VMEAN(4), S(30), V(201,6),
     C                PDEV(200), X1L(200), X2L(200), X3L(200), YL(200),
     C                YC(200), YDEV(200)
      COMMON          ISC(4), JX(6), FMT(18), TITLE(16), IND, IPAGE, IEQ,
     C                IORD, NID, NRUN
      DIMENSION       ID1(201), ID2(201), Y(201), X1(201), X2(201),
     C                X3(201), F(1)
      EQUIVALENCE     (F(1), A), (ID1(1), V(1,1)), (ID2(1), V(1,2)),
     C                (Y(1), V(1,3)), (X1(1), V(1,4)), (X2(1), V(1,5)),
     C                (X3(1), V(1,6)), (JX(1), J1), (JX(2), J2),
     C                (JX(3), J3), (JX(4), J4), (JX(5), J5), (JX(6), J6)
C
C     SUBROUTINE FOR ORDERING DATA AND OBTAINING VARIOUS SUMS AND
C      STANDARD DEVIATIONS OF INPUT DATA
C
   10 IF (IORD .EQ. 0) GO TO 50
C     ORDER THE DATA FROM LOW TO HIGH VALUES OF Y.
      NK = N - 1
      DO 40 I = 1, NK
      IN = I + 1
      DO 30 J = IN, N
      IF (Y(I) .LE. Y(J)) GO TO 30
      DO 20 K = 1, 6
      TEMP = V(I,K)
      V(I,K) = V(J,K)
      V(J,K) = TEMP
   20 CONTINUE
   30 CONTINUE
   40 CONTINUE
C     OBTAIN VARIOUS SUMS OF INPUT DATA.
   50 DO 60 I = 1, N
      YL(I) = ALOG(Y(I))
      X1L(I) = ALOG(X1(I))
      X1SQ = X1(I) * X1(I)
      S(1) = S(1) + Y(I)
      S(2) = S(2) + X1(I)
      S(3) = S(3) + X2(I)
      S(4) = S(4) + X3(I)
      S(5) = S(5) + X1SQ
      S(6) = S(6) + (X1(I) * X2(I))
      S(7) = S(7) + (X1(I) * X3(I))
      S(8) = S(8) + (X2(I) * X2(I))
      S(9) = S(9) + (X2(I) * X3(I))
       S(10) = S(10) + (X3(I) * X3(I))
       S(11) = S(11) + (X1(I) * Y(I))
       S(12) = S(12) + (X2(I) * Y(I))
       S(13) = S(13) + (X3(I) * Y(I))
      IF (IEQ .EQ. 1) GO TO 60
      S(14) = S(14) + YL(I)
      S(15) = S(15) + X1L(I)
```

```
      S(16) = S(16) + (XI(I) * YL(I))
      S(17) = S(17) + (XIL(I) * XIL(I))
      IF (IEQ .NE. 2) GO TO 60
      S(18) = S(18) + (XISQ * XI(I))
      S(19) = S(19) + (XISQ * XISQ)
      S(20) = S(20) + (XISQ * Y(I))
   60 CONTINUE
C     CALCULATE MEANS OF INPUT DATA.
      DO 70 I = 1, 4
      VMEAN(I) = S(I)/AN
   70 CONTINUE
C     CALCULATE SUMS OF THE INPUT DATA ABOUT THEIR MEANS.
      DO 90 K  = 1, 4
      DO 80 I  = 1, N
      S(K+20) = S(K+20) + ((V(I,K+2) - VMEAN(K))**2)
   80 CONTINUE
   90 CONTINUE
C     CALCULATE STANDARD DEVIATIONS OF INPUT DATA.
      DO 100 K = 1, 4
      SDEV(K) = SQRT(S(K+20)/AN1)
  100 CONTINUE
      RETURN
      END
```

```
$IBFTC LINE
      SUBROUTINE LINE
      COMMON        A, A1, B, B1, C, C1, D, D1, AH, BH, CH, DH, AN, AN1,
     C              AN2, DF1, DF2, DFT, FVALUE, CD, CV, R, PDEVM, SEY,
     C              XV, YV, YDEVSQ, EA
      COMMON        IA, IDISK, IERR, IFV, ISOLVE, ISCI, N, NIV, NOTE, NP
      COMMON        H(4,4), T(4), SDEV(4), VMEAN(4), S(30), V(201,6),
     C              PDEV(200), X1L(200), X2L(200), X3L(200), YL(200),
     C              YC(200), YDEV(200)
      COMMON        ISC(4), JX(6), FMT(16), TITLE(16), IND, IPAGE, IEQ,
     C              IORD, NID, NRUN
      DIMENSION     ID1(201), ID2(201), Y(201), X1(201), X2(201),
     C              X3(201), F(1)
      EQUIVALENCE   (F(1), A), (ID1(1), V(1,1)), (ID2(1), V(1,2)),
     C              (Y(1), V(1,3)), (X1(1), V(1,4)), (X2(1), V(1,5)),
     C              (X3(1), V(1,6)), (JX(1), J1), (JX(2), J2),
     C              (JX(3), J3), (JX(4), J4), (JX(5), J5), (JX(6), J6)
C
C     SUBROUTINE FOR DETERMINING LEAST-SQUARES SOLUTIONS OF PARAMETERS
C      FOR LINEAR EQUATIONS OF FORM
C
C                         Y = A + (B*X1) + (C*X2) + (D*X3),
C
C      WHERE A MAY BE SPECIFIED
C
C     DETERMINE VALUES FOR COEFFICIENTS OF LINEAR EQUATIONS TO BE
C      SOLVED.
   10 IF (IA .EQ. 1) GO TO 20
      H(1,1) = AN
      H(1,2) = S(2)
      H(1,3) = S(3)
      H(1,4) = S(4)
      H(2,2) = S(5)
      H(2,3) = S(6)
      H(2,4) = S(7)
      H(3,3) = S(8)
      H(3,4) = S(9)
      H(4,4) = S(10)
      H(2,1) = H(1,2)
      H(3,1) = H(1,3)
      H(3,2) = H(2,3)
      H(4,1) = H(1,4)
      H(4,2) = H(2,4)
      H(4,3) = H(3,4)
      T(1) = S(1)
      T(2) = S(11)
      T(3) = S(12)
      T(4) = S(13)
C     SOLVE FOR PARAMETERS.
      CALL SOLVE
      IF (IERR .EQ. 1) RETURN
      A = AH
      B = BH
      C = CH
      D = DH
      GO TO 30
```

```
   20 H(1,1) = S(5)
      H(1,2) = S(6)
      H(1,3) = S(7)
      H(2,1) = H(1,2)
      H(2,2) = S(8)
      H(2,3) = S(9)
      H(3,1) = H(1,3)
      H(3,2) = H(2,3)
      H(3,3) = S(10)
      T(1) = S(11) - (A * S(2))
      T(2) = S(12) - (A * S(3))
      T(3) = S(13) - (A * S(4))
C     SOLVE FOR PARAMETERS.
      CALL SOLVE
      IF (IERR .EQ. 1) RETURN
      B = AH
      C = BH
      D = CH
C     COMPUTED VALUES OF Y AND Y RESIDUALS
   30 DO 40 I = 1, N
      YC(I) = A + (B * X1(I)) + (C * X2(I)) + (D * X3(I))
      YDEV(I) = Y(I) - YC(I)
   40 CONTINUE
      RETURN
      END
```

```
$IBFTC PARA
      SUBROUTINE PARA
      COMMON        A, A1, B, B1, C, C1, D, D1, AH, BH, CH, DH, AN, AN1,
     C              AN2, DF1, DF2, DFT, FVALUE, CD, CV, R, PDEVM, SEY,
     C              XV, YV, YDEVSQ, EA
      COMMON        IA, IDISK, IERR, IFV, ISOLVE, ISC1, N, NIV, NOTE, NP
      COMMON        H(4,4), T(4), SDEV(4), VMEAN(4), S(30), V(201,6),
     C              PDEV(200), X1L(200), X2L(200), X3L(200), YL(200),
     C              YC(200), YDEV(200)
      COMMON        ISC(4), JX(6), FMT(18), TITLE(16), IND, IPAGE, IEQ,
     C              IORD, NID, NRUN
      DIMENSION     ID1(201), ID2(201), Y(201), X1(201), X2(201),
     C              X3(201), F(1)
      EQUIVALENCE   (F(1), A), (ID1(1), V(1,1)), (ID2(1), V(1,2)),
     C              (Y(1), V(1,3)), (X1(1), V(1,4)), (X2(1), V(1,5)),
     C              (X3(1), V(1,6)), (JX(1), J1), (JX(2), J2),
     C              (JX(3), J3), (JX(4), J4), (JX(5), J5), (JX(6), J6)
C
C     SUBROUTINE FOR DETERMINING LEAST-SQUARES SOLUTIONS OF PARAMETERS
C      FOR PARABOLIC EQUATIONS OF FORM
C
C                         Y = A + (B*X) + (C*(X**2)),
C
C      WHERE A MAY BE SPECIFIED
C
   10 IF (IA .EQ. 1) GO TO 20
      H(1,1) = AN
      H(1,2) = S(2)
      H(1,3) = S(5)
      H(2,1) = H(1,2)
      H(2,2) = S(5)
      H(2,3) = S(18)
      H(3,1) = H(1,3)
      H(3,2) = H(2,3)
      H(3,3) = S(19)
      T(1) = S(1)
      T(2) = S(11)
      T(3) = S(20)
C     SOLVE FOR PARAMETERS.
      CALL SOLVE
      IF (IERR .EQ. 1) RETURN
      A = AH
      B = BH
      C = CH
      GO TO 30
   20 H(1,1) = S(5)
      H(1,2) = S(18)
      H(2,1) = H(1,2)
      H(2,2) = S(19)
      T(1) = S(11) - (A * S(2))
      T(2) = S(20) - (A * S(5))
C     SOLVE FOR PARAMETERS.
      CALL SOLVE
      IF (IERR .EQ. 1) RETURN
      B = AH
      C = BH
```

```
C       COMPUTED VALUES OF Y AND Y RESIDUALS
   30   DO 40 I = 1, N
        YC(I) = A + (B * X1(I)) + (C * (X1(I) * X1(I)))
        YDEV(I) = Y(I) - YC(I)
   40   CONTINUE
C       CALCULATE COORDINATES OF VERTEX POINT.
        XV = -B/(2.0 * C)
        YV = A + (B * XV) + (C * XV * XV)
        RETURN
        END
```

```
$IBFTC POWR
      SUBROUTINE POWR
      COMMON        A, A1, B, B1, C, C1, D, D1, AH, BH, CH, DH, AN, AN1,
     C              AN2, DF1, DF2, DFT, FVALUE, CD, CV, R, PDEVM, SEY,
     C              XV, YV, YDEVSQ, EA
      COMMON        IA, IDISK, IERR, IFV, ISOLVE, ISC1, N, NIV, NOTE, NP
      COMMON        H(4,4), T(4), SDEV(4), VMEAN(4), S(30), V(201,6),
     C              PDEV(200), X1L(200), X2L(200), X3L(200), YL(200),
     C              YC(200), YDEV(200)
      COMMON        ISC(4), JX(6), FMT(18), TITLE(16), IND, IPAGE, IEQ,
     C              IORG, NID, NRUN
      DIMENSION     ID1(201), ID2(201), Y(201), X1(201), X2(201),
     C              X3(201), F(1)
      EQUIVALENCE  (F(1), A), (ID1(1), V(1,1)), (ID2(1), V(1,2)),
     C              (Y(1), V(1,3)), (X1(1), V(1,4)), (X2(1), V(1,5)),
     C              (X3(1), V(1,6)), (JX(1), J1), (JX(2), J2),
     C              (JX(3), J3), (JX(4)  J4), (JX(5), J5), (JX(6), J6)
C
C     SUBROUTINE FOR DETERMINING LEAST-SQUARES SOLUTIONS OF PARAMETERS
C      FOR POWER EQUATIONS OF FORM
C
C                         Y = A * (X1**B) * (X2**C) * (X3**D)
C
C     SET SUBROUTINE INDICATOR TO 2.
   10 IND = 2
      DO 20 I = 1, N
C     SET ALL VALUES OF EITHER THIRD (X3) OR SECOND AND THIRD (X2, X3)
C      INDEPENDENT VARIABLES TO 1 IF NOT BEING CONSIDERED FOR THIS RUN.
      IF (NIV .LT. 3) X3(I) = 1.0
      IF (NIV .LT. 2) X2(I) = 1.0
      X2L(I) = ALOG(X2(I))
      X3L(I) = ALOG(X3(I))
      H(1,2) = H(1,2) + X1L(I)
      H(1,3) = H(1,3) + X2L(I)
      H(1,4) = H(1,4) + X3L(I)
      H(2,2) = H(2,2) + (X1L(I) * X1L(I))
      H(2,3) = H(2,3) + (X1L(I) * X2L(I))
      H(2,4) = H(2,4) + (X1L(I) * X3L(I))
      H(3,3) = H(3,3) + (X2L(I) * X2L(I))
      H(3,4) = H(3,4) + (X2L(I) * X3L(I))
      H(4,4) = H(4,4) + (X3L(I) * X3L(I))
      T(2) = T(2) + (X1L(I) * YL(I))
      T(3) = T(3) + (X2L(I) * YL(I))
      T(4) = T(4) + (X3L(I) * YL(I))
   20 CONTINUE
      H(1,1) = AN
      H(2,1) = H(1,2)
      H(3,1) = H(1,3)
      H(3,2) = H(2,3)
      H(4,1) = H(1,4)
      H(4,2) = H(2,4)
      H(4,3) = H(3,4)
      T(1) = S(14)
C     FIRST, DETERMINE LEAST-SQUARES SOLUTIONS OF PARAMTERS A1, B1, C1,
C      D1 FOR LOGARITHMIC FORM
C
```

```
C          LN(Y) = LN(A1) + (B1 * LN(X1)) + (C1 * LN(X2)) + (D1 * LN(X3))
C
      CALL SOLVE
      IF (IERR .EQ. 1) RETURN
      IF (AH .GT. 88.) GO TO 30
      A1 = EXP(AH)
      B1 = BH
      C1 = CH
      D1 = DH
C     DETERMINE LEAST-SQUARES SOLUTIONS OF PARAMETERS A, B, C, D
C      FOR EXPONENTIAL FORM.
      CALL ITER
      RETURN
C     WRITE ERROR MESSAGE.
   30 WRITE (6, 40) AH
   40 FORMAT (1H0//, 10X,       86HTHE EXPONENT AH IS GREATER THAN 88. IN
     C SUBROUTINE POWR.  THIS RUN HAS BEEN TERMINATED. /// 10X,
     C 5HAH = , F15.7 )
      IERR = 1
      RETURN
      END
```

```
$IBFTC ASYM
      SUBROUTINE ASYM
      COMMON        A, A1, B, B1, C, C1, D, D1, AH, BH, CH, DH, AN, AN1,
     C              AN2, DF1, DF2, DFT, FVALUE, CD, CV, R, PDEVM, SEY,
     C              XV, YV, YDEVSQ, EA
      COMMON        IA, IDISK, IERR, IFV, ISOLVE, ISC1, N, NIV, NOTE, NP
      COMMON        H(4,4), T(4), SDEV(4), VMEAN(4), S(30), V(201,6),
     C              PDEV(200), X1L(200), X2L(200), X3L(200), YL(200),
     C              YC(200), YDEV(200)
      COMMON        ISC(4), JX(6), FMT(18), TITLE(16), IND, IPAGE, IEQ,
     C              IORD, NID, NRUN
      DIMENSION     ID1(201), ID2(201), Y(201), X1(201), X2(201),
     C              X3(201), F(1)
      EQUIVALENCE   (F(1), A), (ID1(1), V(1,1)), (ID2(1), V(1,2)),
     C              (Y(1), V(1,3)), (X1(1), V(1,4)), (X2(1), V(1,5)),
     C              (X3(1), V(1,6)), (JX(1), J1), (JX(2), J2),
     C              (JX(3), J3), (JX(4), J4), (JX(5), J5), (JX(6), J6)
      DIMENSION     SUM(6), QSAVE(3)
C
C     SUBROUTINE FOR DETERMINING LEAST-SQUARES SOLUTIONS OF PARAMETERS
C      FOR ASYMPTOTIC-POWER EQUATIONS OF FORM
C
C                           Y = (A * (X**B)) + C
C
C     SET INITIAL VALUE OF B TO -4.01.
   10 B1 = -4.01
C     SET FIRST-ITERATION DESIGNATOR TO 1.
      ITERS = 1
   20 DO 190 K = 1, 160
      ASTORE = 0.
      BSTORE = 0.
      CSTORE = 0.
C     SET B INCREMENT INITIALLY TO 0.05
   30 DB = 0.05
C     STEP INITIAL B VALUE BY DB INCREMENT.
      B1 = B1 + DB
      IF (K .NE. 81) GO TO 40
      ITERS = 1
      B1 = 0.01
   40 B = B1
   50 DO 60 I = 1, 6
      SUM(I) = 0.
   60 CONTINUE
   70 DO 80 I = 1, N
      XP = X1(I)**B
      XPSQ = XP * XP
      XP1 = XP * X1L(I)
      SUM(1) = SUM(1) + XP
      SUM(2) = SUM(2) + XPSQ
      SUM(3) = SUM(3) + XP1
      SUM(4) = SUM(4) + (XPSQ * X1L(I))
      SUM(5) = SUM(5) + (Y(I) * XP1)
      SUM(6) = SUM(6) + ((Y(I) - VMEAN(1)) * XP)
   80 CONTINUE
      A = SUM(6)/(SUM(2) - (SUM(1) * SUM(1)/AN))
      IF (IA .EQ. 0) C = (S(1) - (A * SUM(1)))/AN
```

```
      G = SUM(5) - (A * SUM(4)) - (C * SUM(3))
      IF (K .EQ. 1) QSAVE(1) = S(21) - (A * SUM(6))
      IF (K .EQ. 160) QSAVE(3) = S(21) - (A * SUM(6))
      IF (ITERS .EQ. 2) GO TO 120
      IF (G) 90, 170, 100
   90 M = -1
      GO TO 110
  100 M = 1
  110 ITERS = 2
      GO TO 190
  120 IF (M .GT. 0) GO TO 130
      IF (G) 150, 170, 140
  130 IF (G) 140, 170, 150
  140 B = B - (DB * 0.5)
      GO TO 160
  150 IF (DB .GT. 0.04) GO TO 190
      B = B + (DB * 0.5)
  160 DDA = ABS(A - ASTORE)
      DDB = ABS(B - BSTORE)
      DDC = ABS(C - CSTORE)
      IF (DDA .LE. 1.0E-08 .AND. DDB .LE. 1.0E-08 .AND. DDC .LE.1.0E-08)
     C GO TO 170
      ASTORE = A
      BSTORE = B
      CSTORE = C
      DB = DB * 0.5
      GO TO 50
C     USE NEW VARIABLES FOR TEMPORARY SOLUTION.
  170 AA = A
      BB = B
      CC = C
C     SUM OF SQUARES OF Y RESIDUALS
      YDEVSQ = S(21) - (AA * SUM(6))
      IF (ISOLVE .EQ. 0) GO TO 180
      IF (YDEVSQ .LE. QSAVE(2)) GO TO 180
      AA = ASAVE
      BB = BSAVE
      CC = CSAVE
      YDEVSQ = QSAVE(2)
C     STORE PARAMETER VALUES AND SUM OF SQUARES OF Y RESIDUALS.
  180 ASAVE = AA
      BSAVE = BB
      CSAVE = CC
      QSAVE(2) = YDEVSQ
C     SET FIRST-ITERATION DESIGNATOR TO 1.
      ITERS = 1
C     SET SOLUTION DESIGNATOR TO 1.
      ISOLVE = 1
  190 CONTINUE
C     IF A UNIQUE SOLUTION FOR B DOES NOT EXIST IN THE SPECIFIED RANGE,
C      PRINT A MESSAGE RELATING TO THAT FACT.
      IF (ISOLVE .EQ. 1 .AND. YDEVSQ .LT. QSAVE(1) .AND.
     C YDEVSQ .LT. QSAVE(3)) GO TO 210
      WRITE (6, 200)
  200 FORMAT (1H0//, 9X, 75HNO SOLUTION HAS BEEN FOUND FOR THIS PROBLEM
     C IN THE RANGE OF -4 TO +4 FOR B. )
```

```
      IERR = 1
      RETURN
C     RESTORE VARIABLES TO ORIGINALS
  210 A = AA
      B = BB
      C = CC
C     CALCULATED VALUES OF Y AND Y RESIDUALS
      DO 220 I = 1, N
      YC(I) = (A * (X1(I)**B)) + C
      YDEV(I) = Y(I) - YC(I)
  220 CONTINUE
      RETURN
      END
```

```
$IBFTC EXPO
      SUBROUTINE EXPO
      COMMON          A, A1, B, B1, C, C1, D, D1, AH, BH, CH, DH, AN, AN1,
     C                AN2, DF1, DF2,   T, FVALUE, CD, CV, R, PDEVM, SEY,
     C                XV, YV, YDEVSQ, EA
      COMMON          IA, IDISK, IERR, IFV, ISOLVE, ISC1, N, NIV, NOTE, NP
      COMMON          H(4,4), T(4), SDEV(4), VMEAN(4), S(30), V(201,6),
     C                PDEV(200), X1L(200), X2L(200), X3L(200), YL(200),
     C                YC(200), YDEV(200)
      COMMON          ISC(4), JX(6), FMT(18), TITLE(16), IND, IPAGE, IEQ,
     C                IORD, NID, NRUN
      DIMENSION       ID1(201), ID2(201), Y(201), X1(201), X2(201),
     C                X3(201), F(1)
      EQUIVALENCE     (F(1), A), (ID1(1), V(1,1)), (ID2(1), V(1,2)),
     C                (Y(1), V(1,3)), (X1(1), V(1,4)), (X2(1), V(1,5)),
     C                (X3(1), V(1,6)), (JX(1), J1), (JX(2), J2),
     C                (JX(3), J3), (JX(4), J4), (JX(5), J5), (JX(6), J6)
C
C     SUBROUTINE FOR DETERMINING LEAST-SQUARES SOLUTIONS OF PARAMETERS
C      FOR EXPONENTIAL EQUATIONS OF FORM
C
C                          Y = EXP(A + (B * X))
C
C     FIRST, OBTAIN LEAST-SQUARES SOLUTIONS OF PARAMETERS A1, B1 FOR
C      SEMI-LOGARITHMIC FORM
C
C                          LN(Y) =  A1 + (B1 * X)
C
C
C     SET SUBROUTINE INDICATOR TO 2.
   10 IND = 2
C     CALCULATE DENOMINATOR OF A1 TERM.
      DENOM = (AN * S(5)) - (S(2) * S(2))
C     CHECK FOR ZERO DENOMINATORS.
      IF (DENOM .EQ. 0. .OR. S(2) .EQ. 0.) GO TO 20
C     CALCULATE NUMERATOR OF A1 TERM.
      ANUM = (S(5) * S(14)) - (S(2) * S(16))
C     CALCULATE A1 TERM.
      A1 = ANUM/DENOM
C     CALCULATE B1 TERM.
      B1 = (S(14) - (AN * A1))/S(2)
C     DETERMINE LEAST-SQUARES SOLUTIONS OF PARAMETERS A, B FOR
C      EXPONENTIAL FORM
      CALL ITER
      RETURN
C     WRITE ERROR MESSAGE.
   20 WRITE (6, 30) DENOM, S(2), AN
   30 FORMAT (1H0//,10X,111HA SOLUTION CANNOT BE OBTAINED BECAUSE A ZERO
     C DENOMINATOR EXISTS IN ONE OR MORE OF THE TERMS (SUBROUTINE EXPO).
     C /// 10X, 8HDENOM = , F15.7 / 10X, 8HS(2)  = , F15.7 / 10X,
     C  8HAN    = , F15.7 )
      IERR = 1
      RETURN
      END
```

```
$IBFTC SOLVE
      SUBROUTINE SOLVE
      COMMON        A, A1, B, B1, C, C1, D, D1, AH, BH, CH, DH, AN, AN1,
     C              AN2, DF1, DF2, DFT, FVALUE, CO, CV, R, PDEVM, SEY,
     C              XV, YV, YDEVSQ, EA
      COMMON        IA, IDISK, IERR, IFV, ISOLVE, ISC1, N, NIV, NOTE, NP
      COMMON        H(4,4), T(4), SDEV(4), VMEAN(4), S(30), V(201,6),
     C              PDEV(200), X1L(200), X2L(200), X3L(200), YL(200),
     C              YC(200), YDEV(200)
      COMMON        ISC(4), JX(6), FMT(18), TITLE(16), IND, IPAGE, IEQ,
     C              IORD, NIC, NRUN
      DIMENSION     ID1(201), ID2(201), Y(201), X1(201), X2(201),
     C              X3(201), F(1)
      DIMENSION     U(12), W(20)
      EQUIVALENCE   (F(1), A), (ID1(1), V(1,1)), (ID2(1), V(1,2)),
     C              (Y(1), V(1,3)), (X1(1), V(1,4)), (X2(1), V(1,5)),
     C              (X3(1), V(1,6)), (JX(1), J1), (JX(2), J2),
     C              (JX(3), J3), (JX(4), J4), (JX(5), J5), (JX(6), J6)
C
C     SUBROUTINE FOR SOLVING SIMULTANEOUS EQUATIONS
C
      GO TO (20, 30, 40, 50), NP
   20 IF (H(1,1) .EQ. 0.) GO TO 60
      AH = T(1)/H(1,1)
      RETURN
   30 DENOM = (H(1,1)/H(1,2)) - (H(2,1)/H(2,2))
      IF (DENOM .EQ. 0.) GO TO 60
      ANUM = (T(1)/H(1,2)) - (T(2)/H(2,2))
      AH = ANUM/DENOM
      BH = (T(1) - (AH * H(1,1)))/H(1,2)
      RETURN
   40 U(1) = (H(1,1) * H(2,3)) - (H(2,1) * H(1,3))
      U(2) = (H(1,2) * H(2,3)) - (H(2,2) * H(1,3))
      U(3) = (T(1) * H(2,3)) - (T(2) * H(1,3))
      U(4) = (H(2,1) * H(3,3)) - (H(3,1) * H(2,3))
      U(5) = (H(2,2) * H(3,3)) - (H(3,2) * H(2,3))
      U(6) = (T(2) * H(3,3)) - (T(3) * H(2,3))
      DENOM = (U(1)/U(2)) - (U(4)/U(5))
      IF (DENOM .EQ. 0.) GO TO 60
      ANUM = (U(3)/U(2)) - (U(6)/U(5))
      AH = ANUM/DENOM
      BH = (U(3) - (AH * U(1)))/U(2)
      CH = (T(1) - (AH * H(1,1)) - (BH * H(1,2)))/H(1,3)
      RETURN
   50 U(1) = (H(1,1) * H(2,4)) - (H(2,1) * H(1,4))
      U(2) = (H(1,2) * H(2,4)) - (H(2,2) * H(1,4))
      U(3) = (H(1,3) * H(2,4)) - (H(2,3) * H(1,4))
      U(4) = (T(1)   * H(2,4))   (T(2)   * H(1,4))
      U(5) = (H(1,3) * H(4,4)) - (H(1,4) * H(3,4))
      U(6) = (H(2,3) * H(4,4)) - (H(2,4) * H(3,4))
      U(7) = (H(3,3) * H(4,4)) - (H(3,4) * H(3,4))
      U(8) = (T(3)   * H(4,4)) - (T(4)   * H(3,4))
      U(9) = (H(1,2) * H(4,4)) - (H(1,4) * H(2,4))
      U(10) = (H(2,2) * H(4,4)) - (H(2,4) * H(2,4))
      U(11) = (H(3,2) * H(4,4)) - (H(3,4) * H(2,4))
      U(12) = (T(2)   * H(4,4)) - (T(4)   * H(2,4))
```

```
      W(1) = U(7)/U(3)
      W(2) = U(7)/U(6)
      W(3) = U(11)/U(7)
      W(4) = (U(4)/U(10)) * (U(7)/U(2))
      W(5) = W(1) - (U(8)/U(4))
      W(6) = W(3) - (U(10)/U(6))
      W(7) = W(2) * (U(8)/U(10))
      W(8) = W(1) - (U(6)/U(2))
      W(9) = W(3) - (U(12)/U(8))
      W(10) = U(1)/U(2)
      W(11) = W(1) - (U(5)/U(1))
      W(12) = (U(11)/U(10)) - W(2)
      W(13) = (U(5)/U(6)) * (U(7)/U(10))
      W(14) = W(1) - (U(6)/U(2))
      W(15) = W(3) - (U(9)/U(5))
      W(16) = W(4) * W(5) * W(6)
      W(17) = W(7) * W(8) * W(9)
      DENOM = (W(10) * W(11) * W(12)) - (W(13) * W(14) * W(15))
      IF (DENOM .EQ. 0.) GO TO 60
      ANUM = W(16) - W(17)
      AH = ANUM/DENOM
      DENOM1 = (U(2)/U(3)) - (U(6)/U(7))
      IF (DENOM1 .EQ. 0.) GO TO 60
      W(18) = (U(4)/U(3)) - (U(8)/U(7))
      W(19) = (U(1)/U(3)) - (U(5)/U(7))
      ANUM1 = W(18) - (AH * W(19))
      BH = ANUM1/DENOM1
      CH = (U(4) - (AH * U(1)) - (BH * U(2)))/U(3)
      DH = (T(1) - (AH * H(1,1)) - (BH * H(1,2)) - (CH * H(1,3)))/H(1,4)
      RETURN
C     ERROR MESSAGE
   60 WRITE (6, 70)
   70 FORMAT (1H0//10X, 97HA ZERO DENOMINATOR EXISTS IN THE CALCULATIONS
     C OF SUBROUTINE SOLVE.  THIS RUN HAS BEEN TERMINATED.)
      GO TO (120, 80, 100), IND
   80 WRITE (6, 90)
   90 FORMAT (1H0/ 10X,  61HNOTE.  SUBROUTINE SOLVE WAS LAST CALLED FROM
     C SUBROUTINE POWR. )
      GO TO 120
  100 WRITE (6, 110)
  110 FORMAT (1H0/ 10X,  61HNOTE.  SUBROUTINE SOLVE WAS LAST CALLED FROM
     C SUBROUTINE ITER. )
  120 IERR = 1
      RETURN
      END
```

```
$IBFTC ITER
      SUBROUTINE ITER
      COMMON       A, A1, B, B1, C, C1, D, D1, AH, BH, CH, DH, AN, AN1,
     C             AN2, DF1, DF2, DFT, FVALUE, CO, CV, R, PDEVM, SEY,
     C             XV, YV, YDEVSQ, EA
      COMMON       IA, IDISK, IERR, IFV, ISOLVE, ISC1, N, NIV, NOTE, NP
      COMMON       H(4,4), T(4), SDEV(4), VMEAN(4), S(30), V(201,6),
     C             PDEV(200), X1L(200), X2L(200), X3L(200), YL(200),
     C             YC(200), YDEV(200)
      COMMON       ISC(4), JX(6), FMT(18), TITLE(16), IND, IPAGE, IEQ,
     C             IORD, NID, NRUN
      DIMENSION    ID1(201), ID2(201), Y(201), X1(201), X2(201),
     C             X3(201), F(1)
      DIMENSION    FP(4), ATEMP(11), BTEMP(11),CTEMP(11),DTEMP(11),Q(11)
      EQUIVALENCE  (F(1), A), (ID1(1), V(1,1)), (ID2(1), V(1,2)),
     C             (Y(1), V(1,3)), (X1(1), V(1,4)), (X2(1), V(1,5)),
     C             (X3(1), V(1,6)), (JX(1), J1), (JX(2), J2),
     C             (JX(3), J3), (JX(4), J4), (JX(5), J5), (JX(6), J6)
C
C     SUBROUTINE FOR DETERMINING LEAST-SQUARES SOLUTIONS OF PARAMETERS
C      FOR NON-LINEAR EQUATIONS WHERE AN ITERATIVE PROCEDURE IS
C      REQUIRED
C
C     STEP SUBROUTINE INDICATOR BY 1.
   10 IND = IND + 1
C     SET INITIAL GUESSES TO LOGARITHMIC SOLUTIONS
      A = A1
      B = B1
      C = C1
      D = D1
   20 DO 200 L = 1, 50
C     CLEAR H AND T MATRICES.
      DO 30 I = 39, 58
      F(I) = 0.
   30 CONTINUE
      IF (IEQ .EQ. 3) GO TO 40
C     CHECK MAGNITUDE OF A FOR EXPONENTIAL CASE.
      IF (A .GT. 88.) GO TO 230
      EA = EXP(A)
   40 DO 100 I = 1, N
      IF (IEQ .EQ. 3) GO TO 50
      E1 = A + (B * X1(I))
      E2 = 2.0 * E1
      E3 = X1L(I) + E2
      E4 = (2.0 * X1L(I)) + E2
C     CHECK MAGNITUDES OF EXPONENTS.
      IF (E2 .GT. 88. .OR. E3 .GT. 88. .OR. E4 .GT. 88.) GO TO 230
C     COMPUTED Y VALUES
      YC(I) = EXP(E1)
C     DIFFERENTIALS OF Y FUNCTION WITH RESPECT TO PARAMETERS A AND B
      FP(1) = YC(I)
      FP(2) = YC(I) * X1(I)
      GO TO 60
C     CHECK MAGNITUDE OF TWICE THE PRODUCT OF EACH PARAMETER (B, C, D)
C      TIMES LOGARITHM OF VALUE OF INDEPENDENT VARIABLE FOR WHICH
C      PARAMETER IS EXPONENT.  IF GREATER THAN 88., PRINT ERROR MESSAGE.
```

```
   50 FAB = 2.0 * B * X1L(I)
      FAC = 2.0 * C * X2L(I)
      FAD = 2.0 * D * X3L(I)
      IF (FAB .GT. 88. .OR. FAC .GT. 88. .OR. FAD .GT. 88.) GO TO 230
C     DIFFERENTIAL OF Y FUNCTION WITH RESPECT TO PARAMETER A
      FP(1) = (X1(I)**B) * (X2(I)**C) * (X3(I)**D)
      YC(I) = A * FP(1)
C     DIFFERENTIALS OF Y FUNCTION WITH RESPECT TO PARAMETERS B, C, AND D
      FP(2) = YC(I) * X1L(I)
      FP(3) = YC(I) * X2L(I)
      FP(4) = YC(I) * X3L(I)
C     Y RESIDUALS
   60 YDEV(I) = Y(I) - YC(I)
      IF (NOTE .EQ. 1) GO TO 100
      IF (ISOLVE .EQ. 0) GO TO 70
      YD = (2.0 * YC(I)) - Y(I)
      IF (YD .LE. 0.) NOTE = 1
      GO TO 100
C     CALCULATE H AND T MATRICES.
   70 DO 90 II = 1, 4
      DO 80 JJ = 1, 4
      H(II,JJ) = H(II,JJ) + (FP(II) * FP(JJ))
   80 CONTINUE
      T(II) = T(II) + (YDEV(I) * FP(II))
   90 CONTINUE
  100 CONTINUE
C     IF A SOLUTION HAS BEEN OBTAINED (ISOLVE = 1), STOP ITERATION AND
C      RETURN.
      IF (ISOLVE .EQ. 1) RETURN
C     SOLVE FOR CORRECTIONS TO PREVIOUS SOLUTIONS.
      CALL SOLVE
      IF (IERR .EQ. 1) RETURN
      DA = AH
      DB = BH
      DC = CH
      DD = DH
C
C     FIND WHICH FRACTIONAL PART OF CORRECTION TERMS, WHEN ADDED TO
C      PARAMETER VALUES, GIVES LOWEST SUM OF SQUARES OF Y RESIDUALS.
C
      TEMP = 1.0
  110 TEMP = 0.1 * TEMP
      DO 120 J = 1, 11
      FI = TEMP * FLOAT(J - 1)
      ATEMP(J) = A + (DA * FI)
      BTEMP(J) = B + (DB * FI)
      CTEMP(J) = C + (DC * FI)
      DTEMP(J) = D + (DD * FI)
      Q(J) = 0.
  120 CONTINUE
      DO 160 J = 1, 11
      DO 150 I = 1, N
      IF (IEQ .EQ. 5) GO TO 130
      YTEMP= ATEMP(J)*(X1(I)**BTEMP(J))*(X2(I)**CTEMP(J))*
     C (X3(I)**DTEMP(J))
      GO TO 140
```

```
  130 YTEMP = EXP(ATEMP(J) + (BTEMP(J) * X1(I)))
  140 YDIF = Y(I) - YTEMP
      Q(J) = Q(J) + (YDIF * YDIF)
  150 CONTINUE
  160 CONTINUE
      YDEVSQ = Q(1)
      LM = 1
      DO 170 J = 2, 11
      IF (YDEVSQ .LE. Q(J)) GO TO 170
      LM = J
      YDEVSQ = Q(J)
  170 CONTINUE
      IF (LM .GT. 1) GO TO 180
      IF (ABS(DA*10.0*TEMP) .GT. 1.0E-08) GO TO 110
      IF (ABS(DB*10.0*TEMP) .GT. 1.0E-08) GO TO 110
      IF (ABS(DC*10.0*TEMP) .GT. 1.0E-08) GO TO 110
      IF (ABS(DD*10.0*TEMP) .GT. 1.0E-08) GO TO 110
  180 DDA = ABS(A - ATEMP(LM))
      DDB = ABS(B - BTEMP(LM))
      DDC = ABS(C - CTEMP(LM))
      DDD = ABS(D - DTEMP(LM))
C     UPDATE THE VALUE OF EACH PARAMETER.
      A = ATEMP(LM)
      B = BTEMP(LM)
      C = CTEMP(LM)
      D = DTEMP(LM)
C     IF A SOLUTION IS OBTAINED, SET SOLUTION DESIGNATOR, ISOLVE, TO 1.
C      (A SOLUTION IS ASSUMED WHEN THE CHANGE IN THE VALUE OF EACH
C      PARAMETER FROM ONE ITERATION TO THE NEXT BECOMES EQUAL TO, OR
C      LESS THAN, 10**(-8)).
  190 IF (DDA .GT. 1.0E-08) GO TO 200
      IF (DDB .GT. 1.0E-08) GO TO 200
      IF (DDC .GT. 1.0E-08) GO TO 200
      IF (DDD .LE. 1.0E-08) ISOLVE = 1
  200 CONTINUE
C
C     ERROR MESSAGES
C
  210 WRITE (6, 220) A1, B1
  220 FORMAT (1H0// 10X,        94HNO SOLUTION HAS BEEN OBTAINED FOR THIS
     C RUN AFTER 50 ITERATIONS.  THIS RUN HAS BEEN TERMINATED. ///
     C 10X, 21HLOGARITHMIC SOLUTIONS / 10X, 5H A = , F14.5 / 10X,
     C 5H B = , F14.5 )
      IF (NIV .EQ. 2) WRITE (6, 222) C1
  222 FORMAT (1H , 9X, 5H C = , F14.5)
      IF (NIV .EQ. 3) WRITE (6, 224) C1, D1
  224 FORMAT (1H , 9X, 5H C = , F14.5 / 10X, 5H D = , F14.5)
      GO TO 250
  230 WRITE (6, 240) A, B, C, D
  240 FORMAT (1H0// 10X,        66HOVERFLOWS EXIST IN SUBROUTINE ITER.
     C THIS RUN HAS BEEN TERMINATED. /// 10X, 4HA = , F14.5 / 10X,
     C 4HB = , F14.5)
      IF (NIV .EQ. 2) WRITE (6, 242) C
  242 FORMAT (1H , 9X, 4HC = , F14.5)
      IF (NIV .EQ. 3) WRITE (6, 244) C, D
  244 FORMAT (1H , 9X, 4HC = , F14.5 / 10X, 4HD = , F14.5)
```

```
250 IERR = 1
    RETURN
    END
```

```
$IBFTC STAT
      SUBROUTINE STAT
      COMMON         A, A1, B, B1, C, C1, D, D1, AH, BH, CH, DH, AN, AN1,
     C               AN2, DF1, DF2, DFT, FVALUE, CD, CV, R, PDEVM, SEY,
     C               XV, YV, YDEVSQ, EA
      COMMON         IA, IDISK, IERR, IFV, ISOLVE, ISC1, N, NIV, NOTE, NP
      COMMON         H(4,4), T(4), SDEV(4), VMEAN(4), S(30), V(201,6),
     C               PDEV(200), X1L(200), X2L(200), X3L(200), YL(200),
     C               YC(200), YDEV(200)
      COMMON         ISC(4), JX(6), FMT(18), TITLE(16), IND, IPAGE, IEQ,
     C               IORD, NID, NRUN
      DIMENSION      ID1(201), ID2(201), Y(201), X1(201), X2(201),
     C               X3(201), F(1)
      EQUIVALENCE    (F(1), A), (ID1(1), V(1,1)), (ID2(1), V(1,2)),
     C               (Y(1), V(1,3)), (X1(1), V(1,4)), (X2(1), V(1,5)),
     C               (X3(1), V(1,6)), (JX(1), J1), (JX(2), J2),
     C               (JX(3), J3), (JX(4), J4), (JX(5), J5), (JX(6), J6)
C
C     SUBROUTINE FOR CALCULATING STATISTICS
C
   10 YDEVSQ = 0.
      DO 20 I = 1, N
C     PERCENT Y DEVIATIONS
      PDEV(I) = (100. * YDEV(I))/Y(I)
C     SUM OF ABSOLUTE PERCENT Y DEVIATIONS
      PDEVM = PDEVM + ABS(PDEV(I))
C     SUM OF SQUARES OF Y RESIDUALS
      YDEVSQ = YDEVSQ + (YDEV(I) * YDEV(I))
C     SUM OF COMPUTED Y VALUES
      S(25) = S(25) + YC(I)
   20 CONTINUE
C     MEAN OF COMPUTED Y VALUES
      YCMEAN = S(25)/AN
C     SUM OF SQUARES OF COMPUTED Y VALUES ABOUT THEIR MEAN
      DO 30 I = 1, N
      S(26) = S(26) + ((YC(I) - YCMEAN)**2)
   30 CONTINUE
C     TOTAL DEGREES OF FREEDOM
      DFT = AN1
C     DEGREES OF FREEDOM ABOUT REGRESSION CURVE
      DF1 = N - NP
C     DEGREES OF FREEDOM DUE TO REGRESSION
      DF2 = DFT - DF1
C     F VALUE
      IFV = 0
      DENOM = YDEVSQ/DF1
      IF (DENOM .NE. 0. .AND. DF2 .NE. 0.) GO TO 40
      IFV = 1
      GO TO 50
   40 ANUM = S(26)/DF2
      FVALUE = ANUM/DENOM
      IF (FVALUE .GE. 1.0E+08) IFV = 1
C     MEAN OF ABSOLUTE PERCENT Y DEVIATIONS
   50 PDEVM = PDEVM/AN
C     STANDARD ERROR OF THE ESTIMATE OF Y (ADJUSTED FOR SAMPLE SIZE)
      SEY = SQRT(YDEVSQ/DF1)
```

```
C     COEFFICIENT OF VARIATION PERCENT
      CV = (100. * SEY)/VMEAN(1)
C     COEFFICIENT OF DETERMINATION (UNADJUSTED FOR SAMPLE SIZE)
      CD = 1.0 - (YDEVSQ/S(21))
C     COEFFICIENT OF CORRELATION (UNADJUSTED) FOR SAMPLE SIZE)
      R = SQRT(CD)
      RETURN
      END
```

```
$IBFTC OUT1
      SUBROUTINE OUT1
      COMMON        A, A1, B, B1, C, C1, D, D1, AH, BH, CH, DH, AN, AN1,
     C              AN2, DF1, DF2, DFT, FVALUE, CD, CV, R, PDEVM, SEY,
     C              XV, YV, YDEVSQ, EA
      COMMON        IA, IDISK, IERR, IFV, ISOLVE, ISC1, N, NIV, NOTE, NP
      COMMON        H(4,4), T(4), SDEV(6), VMEAN(4), S(30), V(201,6),
     C              PDEV(200), X1L(200), X2L(200), X3L(200), YL(200),
     C              YC(200), YDEV(200)
      COMMON        ISC(4), JX(6), FMT(18), TITLE(16), IND, IPAGE, IEQ,
     C              IORD, NID, NRUN
      DIMENSION     ID1(201), ID2(201), Y(201), X1(201), X2(201),
     C              X3(201), F(1)
      EQUIVALENCE   (F(1), A), (ID1(1), V(1,1)), (ID2(1), V(1,2)),
     C              (Y(1), V(1,3)), (X1(1), V(1,4)), (X2(1), V(1,5)),
     C              (X3(1), V(1,6)), (JX(1), J1), (JX(2), J2),
     C              (JX(3), J3), (JX(4), J4), (JX(5), J5), (JX(6), J6)
C
C     SUBROUTINE FOR PRINTING SUMMARY TABLE
C
   10 WRITE (6, 20)
   20 FORMAT (1H0// 59X, 13HSUMMARY TABLE ///)
      NP1 = NP + IA
      IF (IA .EQ. 1) WRITE (6, 25)
   25 FORMAT (1H , 12X,   106HNOTE -- THE STATISTICS CALCULATED FOR THIS
     C RUN ARE NOT COMPARABLE WITH THOSE FOR UNSPECIFIED Y-INTERCEPTS.
     C /// )
   30 WRITE (6, 40) A, B
   40 FORMAT (1H /
     C 30X, 41HA                                       , 10X, F14.5 /
     C 30X, 41HB                                       , 10X, F14.5 )
      GO TO (90, 90, 50, 70), NP1
   50 WRITE (6, 60) C
   60 FORMAT (1H ,
     C 29X, 41HC                                       , 10X, F14.5 )
      IF (IEQ .EQ. 2) WRITE (6, 65) XV, YV
   65 FORMAT (1H /
     C 30X, 41HX COORDINATE OF VERTEX POINT            , 10X, F14.5 /
     C 30X, 41HY COORDINATE OF VERTEX POINT            , 10X, F14.5 )
      GO TO 90
   70 WRITE (6, 80) C, D
   80 FORMAT (1H ,
     C 29X, 41HC                                       , 10X, F14.5 /
     C 30X, 41HD                                       , 10X, F14.5 )
   90 IF (IEQ .NE. 3 .AND. IEQ .NE. 5) GO TO 190
      WRITE (6, 100)
  100 FORMAT (1H / 30X, 21HLOGARITHMIC SOLUTIONS )
      WRITE (6, 110) A1, B1
  110 FORMAT (1H ,
     C 29X, 41H  A                                     , 10X, F14.5 /
     C 30X, 41H  B                                     , 10X, F14.5 )
      GO TO (160, 160, 120, 140), NP1
  120 WRITE (6, 130) C1
  130 FORMAT (1H ,
     C 29X, 41H  C                                     , 10X, F14.5 )
      GO TO 160
```

```
140 WRITE (6, 150) CI, DI
150 FORMAT (1H ,
   C 29X, 41H  C                                              , 10X, F14.5 /
   C 30X, 41H  D                                              , 10X, F14.5 )
160 IF (IEQ .EQ. 5) WRITE (6, 170) EA
170 FORMAT (1H /
   C 30X, 41HY INTERCEPT                                      , 10X, F14.5 )
190 WRITE (6, 200) R, CD, SEY, CV, YDEVSQ, PDEVM
200 FORMAT (1H /
   C 30X, 41HCOEFFICIENT OF CORRELATION (UNADJUSTED)   , 10X, F14.5 /
   C 30X, 41HCOEFFICIENT OF DETERMINATION (UNADJUSTED), 10X, F14.5 /
   C 30X, 46HSTANDARD ERROR OF THE ESTIMATE OF Y (ADJUSTED), 5X,
   C F14.5 /
   C 30X, 41HCOEFFICIENT OF VARIATION (PERCENT)        , 10X, F14.5 /
   C 30X, 41HSUM OF SQUARES OF Y RESIDUALS             , 10X, F14.5 /
   C 30X, 41HMEAN OF ABSOLUTE PERCENT Y DEVIATIONS     , 10X, F14.5 )
    IF (IFV .EQ. 0) WRITE (6, 210) FVALUE
210 FORMAT (1H /
   C 30X, 41HF VALUE                                   , 10X, F14.5 )
    IF (IFV .EQ. 1) WRITE (6, 220)
220 FORMAT (1H /
   C 30X, 41HF VALUE                                   , 16X,
   C 30HEQUAL TO OR GREATER THAN 10**P )
    WRITE (6, 230) DF1, DF2, DFT
230 FORMAT (1H ,
   C 29X, 41H[illegible]EGREES OF FREEDOM ABOUT REGRESSION CURVE, 10X, F14.5 /
   C 30X, 41HDEGREES OF FREEDOM DUE TO REGRESSION      , 10X, F14.5 /
   C 30X, 41HTOTAL DEGREES OF FREEDOM                  , 10X, F14.5 )
    WRITE (6, 235)
235 FORMAT (1H , 30X, 19HMEANS OF INPUT DATA )
    WRITE (6, 240) VMEAN(1), VMEAN(2)
240 FORMAT (1H ,
   C 29X, 41H  Y                                       , 10X, F14.5 /
   C 30X, 41H  X1                                      , 10X, F14.5 )
    GO TO (290, 250, 270), NIV
250 WRITE (6, 260) VMEAN(3)
260 FORMAT (1H ,
   C 29X, 41H  X2                                      , 10X, F14.5 )
    GO TO 290
270 WRITE (6, 280) VMEAN(3), VMEAN(4)
280 FORMAT (1H ,
   C 29X, 41H  X2                                      , 10X, F14.5 /
   C 30X, 41H  X3                                      , 10X, F14.5 )
290 WRITE (6, 295)
295 FORMAT (1H / 30X, 33HSTANDARD DEVIATIONS OF INPUT DATA)
    WRITE (6, 300) SDEV(1), SDEV(2)
300 FORMAT (1H ,
   C 29X, 41H  Y                                       , 10X, F14.5 /
   C 30X, 41H  X1                                      , 10X, F14.5 )
    GO TO (350, 310, 330), NIV
310 WRITE (6, 320) SDEV(3)
320 FORMAT (1H ,
   C 29X, 41H  X2                                      , 10X, F14.5 )
    GO TO 350
330 WRITE (6, 340) SDEV(3), SDEV(4)
340 FORMAT (1H ,
```

```
   C 29X  41H  X2                                   , 10X, F14.5 /
   C 30X, 41H  X3                                   , 10X, F14.5 )
350 WRITE (6, 360) AN
360 FORMAT (1H /
   C 30X, 41HNUMBER OF DATA POINTS                  ,  5X, F14.0 )
    RETURN
    END
```

```
$IBFTC OUT2
      SUBROUTINE OUT2
      COMMON          A, A1, B, B1, C, C1, D, D1, AH, BH, CH, DH, AN, AN1,
     C                AN2, DF1, DF2, DFT, FVALUE, CO, CV, R, PDEVM, SEY,
     C                XV, YV, YDEVSQ, EA
      COMMON          IA, IDISK, IERR, IFV, ISOLVE, ISC1, N, NIV, NOTE, NP
      COMMON          H(4,4), T(4), SDEV(4), VMEAN(4), S(30), V(201,6),
     C                PDEV(200), X1L(200), X2L(200), X3L(200), YL(200),
     C                YC(200), YDEV(200)
      COMMON          ISC(4), JX(6), FMT(18), TITLE(16), IND, IPAGE, IEQ,
     C                ICRD, NID, NRUN
      DIMENSION       ID1(201), ID2(201), Y(201), X1(201), X2(201),
     C                X3(201), F(1)
      EQUIVALENCE     (F(1), A), (ID1(1), V(1,1)), (ID2(1), V(1,2)),
     C                (Y(1), V(1,3)), (X1(1), V(1,4)), (X2(1), V(1,5)),
     C                (X3(1), V(1,6)), (JX(1), J1), (JX(2), J2),
     C                (JX(3), J3), (JX(4), J4), (JX(5), J5), (JX(6), J6)
C
C     SUBROUTINE FOR PRINTING INPUT DATA, CALCULATED Y VALUES,
C      Y RESIDUALS, AND PERCENT Y DEVIATIONS
C
C     PRINT TITLE ON NEW PAGE
   10 WRITE (6, 20) (TITLE(I), I = 1, 16), IPAGE
   20 FORMAT (1H1/ 10X, 16A4, 41X, 5HPAGE , I2 / )
C     STEP PAGE NUMBER BY 1.
      IPAGE = IPAGE + 1
C     SET LINE COUNT TO ZERO.
      LINES = 0
      WRITE (6, 30)
   30 FORMAT (1H0/  49X, 31HCOMPUTED Y VALUES AND RESIDUALS )
      GO TO (40, 100, 140), NIV
   40 WRITE (6, 50)
   50 FORMAT (1H0/ 109X, 7HPERCENT / 14X, 5HLABEL, 17X, 1HY, 17X, 2HX1,
     C 15X, 7HY CALC., 13X, 2(6HY DEV., 13X) / )
      DO 90 I = 1, N
      FI1 = I/5
      FI2 = FLOAT(I)/5.0
      IF (LINES .LT. 40) GO TO 55
      WRITE (6, 20) (TITLE(K), K = 1, 16), IPAGE
      IPAGE = IPAGE + 1
      WRITE (6, 80)
      WRITE (6, 50)
      LINES = 0
   55 WRITE (6, 60) (V(I,K), K = 1, 4), YC(I), YDEV(I), PDEV(I)
   60 FORMAT (1H , 13X, 2A4, 5(5X, F14.5) )
      LINES = LINES + 1
      IF (FI1 .EQ. FI2) WRITE (6, 70)
   70 FORMAT (1H )
   80 FORMAT (1H0/ 44X, 43HCOMPUTED Y VALUES AND RESIDUALS (CONTINUED) )
   90 CONTINUE
      GO TO 180
  100 WRITE (6, 110)
  110 FORMAT (1H0/ 109X, 7HPERCENT / 10X, 5HLABEL. 15X, 1HY, 15X, 2HX1,
     C 15X, 2HX2, 13X, 7HY CALC., 11X, 2(6HY DEV., 11X) / )
      DO 130 I = 1, N
      FI1 = I/5
```

```
      FI2 = FLOAT(I)/5.0
      IF (LINES .LT. 40) GO TO 115
      WRITE (6, 20) (TITLE(K), K = 1, 16), IPAGE
      IPAGE = IPAGE + 1
      WRITE (6, 80)
      WRITE (6, 110)
      LINES = 0
  115 WRITE (6, 120) (V(I,K), K = 1, 5), YC(I), YDEV(I), PDEV(I)
  120 FORMAT (1H ,  9X, 2A4, 6(3X, F14.5) )
      LINES = LINES + 1
      IF (FI1 .EQ. FI2) WRITE (6, 70)
  130 CONTINUE
      GO TO 180
  140 WRITE (6, 150)
  150 FORMAT (1HO/ 121X, 7HPERCENT / 2X, 5HLABEL, 15X, 1HY, 15X, 2HX1,
     C 15X, 2HX2, 15X, 2HX3, 13X, 7HY CALC., 11X, 6HY DEV., 11X,
     C 6HY DEV. / )
      DO 170 I = 1, N
      FI1 = I/5
      FI2 = FLOAT(I)/5.0
      IF (LINES .LT. 40) GO TO 155
      WRITE (6, 20) (TITLE(K), K = 1, 16), IPAGE
      IPAGE = IPAGE + 1
      WRITE (6, 80)
      WRITE (6, 150)
      LINES = 0
  155 WRITE (6, 160) (V(I,K), K = 1, 6), YC(I), YDEV(I), PDEV(I)
  160 FORMAT (1H , 1X, 2A4, 7(3X, F14.5) )
      LINES = LINES + 1
      IF (FI1 .EQ. FI2) WRITE (6, 70)
  170 CONTINUE
  180 IF (NOTE .EQ. 1) WRITE (6, 190)
  190 FORMAT (1HO/, 10X,   109HTHE ABOVE SOLUTION FOR THIS CASE DOES NOT
     C GIVE AN ABSOLUTE MINIMIZATION OF THE SUM OF SQUARES OF Y RESIDUAL
     CS. )
      RETURN
      END
```

DOCUMENT CONTROL DATA

1. ORIGINATING ACTIVITY THE RAND CORPORATION	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

3. REPORT TITLE

CURVES: A FIVE-FUNCTION CURVE-FITTING COMPUTER PROGRAM

4. AUTHOR(S) (Last name, first name, initial)

Boren, Jr., H.E.

5. REPORT DATE	6a. TOTAL No. OF PAGES	6b. No. OF REFS.
December 1968	94	---

7. CONTRACT OR GRANT No.	8. ORIGINATOR'S REPORT No.
F44620-67-C-0045	RM-5762-PR

9a. AVAILABILITY / LIMITATION NOTICES	9b. SPONSORING AGENCY
DDC-1	United States Air Force Project RAND

10. ABSTRACT

Description and listing of an all-FORTRAN IV program that makes least-squares determinations of the parameters of any of five mathematical functions selected by the user, given a set of observations on the dependent and independent variables of interest (up to 200 data points per curve). The functions available are those most commonly used in developing cost estimating relationships: line, parabola, power, asymptotic-power, and exponential. Up to three independent variables may be used for the line and power functions. The Y-intercept may be specified for the line, parabola, or asymptotic-power functions. The program is designed to be user-oriented and easily workable rather than to emphasize computational efficiency. Exact and unique solutions for the line and parabolic functions are obtained by standard algebraic methods. Since the other three choices are not linear in all parameters, they are solved iteratively: the power and exponential by a modified Gauss-Newton iteration, starting from the exact logarithmic solution (as described in RM-4879-PR), and the asymptotic-power function by a special iterative method described in the RM. In addition to directions for program use, the RM includes a discussion of the characteristics of the functions and mathematical considerations involved in nonlinear least-squares solutions.

11. KEY WORDS

Curve fitting
Cost estimating relationships
Computer programs
Statistical methods and processes